JOINT PATH AND RESOURCE SELECTION FOR OBS GRIDS WITH ADAPTIVE OFFSET BASED QOS MECHANISM

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND

ELECTRONICS ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCES

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Mehmet Köseoğlu

September 2007

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Ezhan Karaşan(Supervisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Nail Akar

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. İbrahim Körpeoğlu

Approved for the Institute of Engineering and Sciences:

Prof. Dr. Mehmet Baray
Director of Institute of Engineering and Sciences

ABSTRACT

JOINT PATH AND RESOURCE SELECTION FOR OBS GRIDS WITH ADAPTIVE OFFSET BASED QOS MECHANISM

Mehmet Köseoğlu
M.S. in Electrical and Electronics Engineering
Supervisor: Assoc. Prof. Dr. Ezhan Karaşan
September 2007

It is predicted that grid computing will be available for consumers performing their daily computational needs with the deployment of high bandwidth optical networks. Optical burst switching is a suitable switching technology for this kind of consumer grid networks because of its bandwidth granularity. However, high loss rates inherent in OBS has to be addressed to establish a reliable transmission infrastructure. In this thesis, we propose mechanisms to reduce loss rates in an OBS grid scenario by using network-aware resource selection and adaptive offset determination.

We first propose a congestion-based joint resource and path selection algorithm. We show that path switching and network-aware resource selection can reduce burst loss probability and average completion time of grid jobs compared to the algorithms that are separately selecting paths and grid resources. In addition to joint resource and path selection, we present an adaptive offset algorithm for grid bursts which minimizes the average completion time. We show that the adaptive offset based QoS mechanism significantly reduces the job completion

times by exploiting the trade-off between decreasing loss probability and increasing delay as a result of the extra offset time. *Keywords:* Grid Networks, Optical Burst Switching, Grid Resource Selection, Photonic Grid

ÖZET

OPTİK ÇOĞUŞMA ANAHTARLAMALI GRİD AĞLARINDA BÜTÜNLEŞİK KAYNAK-YOL SEÇİMİ VE UYARLAMALI OFSET TABANLI SERVİS KALİTESİ MEKANİZMASI

Mehmet Köseoğlu
Elektrik ve Elektronik Mühendisliği Bölümü Yüksek Lisans
Tez Yöneticisi: Doç. Dr. Ezhan Karaşan
Eylül 2007

Yüksek bant genişliğine sahip fiber optik ağların yaygınlaşmasıyla tüketicilerin günlük hesaplama ihtiyaçları için grid hizmetlerinden faydalanabilecekleri öngörülmektedir. Optik çoğuşma anahtarlama (OÇA) küçük öğe boyu sayesinde bu gibi kullanıcı grid ağları için uygun bir anahtarlama teknolojisidir. Fakat güvenilir bir iletişim altyapısının kurulabilmesi için OÇA protokolünün doğasından kaynaklanan veri kayıplarının azaltılması gerekmektedir. Bu tezde OÇA anahtarlaması kullanan bir grid ağında ağ-farkında kaynak seçimi algoritması ve uyarlanabilir ofset tesbiti metoduyla kayıp oranlarını azaltan bir mekanizma sunuyoruz.

Öncelikle, sıkışıklık tabanlı bütünleşik kaynak ve yol seçimi algoritması açıklanmıştır. Simulasyonlarımız yol anahtarlama ve ağ-farkında kaynak seçiminin coğuşma kayıp olasılığını ve grid işlerinin ortalama tamamlanma zamanını azalttığını göstermiştir. Bütünleşik kaynak ve yol seçimine ek olarak, grid coğuşmaları için ortalama tamamlanma zamanını azaltan uyarlamalı ofset algoritması sunulmuştur. Kayıp oranlarındaki azalmayla ofsetten kaynaklanan

gecikme arasında bir denge bulan bu algoritmanın grid işlerinin ortalama tamamlanma zamanını önemli ölçüde azalttığı gösterilmiştir.

Anahtar Kelimeler: Grid Ağları, Optik Coğuşma Anahtarlama, Grid Kaynak Seçimi, Fotonik Grid

ACKNOWLEDGMENTS

I would like to express my gratitude to Dr. Ezhan Karaşan not only for his academic supervision but also for his guidance related to my life. I also would like to thank other members of my thesis committee, Dr. Nail Akar and Dr. İbrahim Körpeoğlu, for their useful comments on this thesis.

I am grateful to Hande Doğan for making my life more enjoyable with her presence.

I must acknowledge my friends at EA-226 and EA-228 for providing a friendly atmosphere. I would also like to thank Aselsan Inc. and my colleagues at Aselsan for their support during the first years of my studies.

Last but not least, I am also thankful to Sami Ezercan for his long lasting friendship.

Contents

List of Figures

Dedicated to my parents ...

Chapter 1

INTRODUCTION

Computing power of processors have increased dramatically since the invention of the computer but individual computers are still inadequate for solving large-scale problems. Some scientific applications such as particle physics experiments performed at the Large Hadron Collider at CERN [1] require enormous processing power and storage capacity which cannot be satisfied using a single supercomputer.

Clustering individual computers is a solution to this problem. In a computer cluster, the computers are connected through a fast local area network and achieve faster execution by running different parts of a single problem in parallel. However, a high performance computing cluster is an expensive investment requiring dedicated servers in a single location and many institutions cannot afford this investment for specific problems.

With the widespread usage of the Internet, applications which make use of the idle computing power of personal computers distributed around the world are developed for scientific research. A well known example is the SETI@home project [2] which analyzes radio transmissions seeking evidence of extra-terrestrial life. The project has over 3 million participants who let their computers work for the

project when they are not using them. The project has a distributed processing power over 200 Teraflops per second which is about the performance of the largest supercomputer in the world.

Beyond Internet computing, institutions can perform resource sharing in a larger scale using higher bandwidth communication networks. With the development of standards providing flexible and secure resource sharing, participation of users with different characteristics running diverse applications becomes possible. This new paradigm is called grid computing [3]. The name of the grid computing comes from the electrical grid, in which users get high quality service in an on demand basis from a resource pool to which several resources are contributing. This electrical grid represents an ideal for grid computing because of its transparency and ease of use and the grid community is working towards this ideal by defining standards and protocol.

The early examples of grid computing are created for the heavy computing needs of advanced scientific problems which cannot be solved locally. These problems involve joint studies of several institutions and, for that reason, the emphasis was on the collaboration of grid resources. The institutions or companies which join and share resources in a collaborative grid are called virtual organizations which have different local administrative policies but collaborate using a uniform interface through the grid.

In contrast to organization-based collaborative grids, the individual-based consumer grid concept is proposed by [4]. In a consumer grid, the resources are used by consumers in a commercial basis not in a collaborative manner. The users of the consumer grid do not necessarily have the same goal and use resources by purchasing them. The supply and demand of computing resources create a computational market where the price of computation is determined dynamically.

Geographical separation of resources in a grid computing environment makes the networking issues important for grid computing in contrast to computer clusters [5]. In computer clusters, the network parameters are usually neglected because of the high bandwidth available inside the organization and because no data transfer is needed to a distant place. However, in grid computing, the duration and quality of service of the data transmission becomes important because of the geographical separation. Without a reliable medium of communication between distant resources collaboration between separated resources is not possible.

As the bandwidth requirements of grid applications increase, optical networks are now being considered as the network infrastructure of grids. For example, the aforementioned particle physics experiments generate terabytes of data which cannot be processed or stored by using only local resources. For that reason, the data has to be carried over long distances to be processed by distributed resources. Deployment of optical networks becomes a necessity for this kind of high bandwidth applications [6].

There are also more potential applications of high bandwidth grid computing which require interaction between the client and the server such as remote rendering. Remote rendering is the rendering of computer generated graphics at a remote location. The reason for such a system is that many applications require very high computing power to render complex graphics and without a local rendering facility, it is impossible to process such graphics. However, with the further deployment optical networks and improvement of QoS guarantees, it is possible to get service from a remote rendering facility. An experiment of remote rendering over an intercontinental optical network is explained in [7]. This experiment shows that remote rendering is possible rate between continents while maintaining interactivity.

In high bandwidth grid computing and distributed peer to peer computing, the user should have a large amount of control over the network resources such as being able to initiate connections, allocate bandwidth, etc. This kind of control mechanism is very different than the traditional telecommunications model, where the service providers has control over the core network and the customers request service from providers without knowing the details of the core network.

There are a number of switching choices for data transmission for a consumer controlled optical network. Wavelength switching and optical burst switching are the most commonly referenced switching mechanisms for optical networks. With the advance of the optical technologies, packet switching could also be possible for optical networks.

Wavelength switching is the switching of wavelengths to different paths to create a virtual circuit for data transmission. For long lasting high bandwidth data streams, wavelength switching is a suitable technology since it provides dedicated wavelengths and guarantee quality of service. If the consumers can dynamically establish lightpaths, it is suitable for non-interactive high bandwidth grid applications such as particle physics experiments.

A novel switching paradigm called Optical Burst Switching(OBS) for optical networks is proposed in [8] because the bandwidth granularity of wavelength switching is very high and it is not suitable for applications with smaller data sizes. OBS lies in between wavelength switching and optical packet switching in terms of data granularity. Unlike packet switching, it does not require buffering so it is a more practical technology for the near future. In OBS, a control packet is sent before the optical data, which is called a burst, to configure the switches along the lightpath. After an offset time, the optical burst is sent without waiting for an acknowledgment of reservation. This one-way reservation makes fast lightpath setup possible and the relatively small sizes of bursts provide

bandwidth granularity and, for that reason, OBS is more suitable for interactive high bandwidth grid applications.

Also, OBS technology is considered as a promising technology for grids because of the following reasons [9]:

- Mapping between grid bursts and grid jobs: It can be possible to map grid jobs to grid bursts one-to-one, so the grid jobs can be effectively transmitted using the bandwidth granularity of OBS.
- Separation of control and data plane: This makes consumer initiated lightpath setup possible and allows all-optical data transmission.
- Processing of control packets at each node: It is possible to integrate grid level functionalities to the OBS control plane such as resource discovery using intelligent routers.

Despite these advantages, there are several problems with OBS for grid computing. Although OBS allows fast transport of bursts, burst contentions in the core network occurs when the reservation attempt by the burst control packet is not successful if the capacity of a link is fully occupied by other bursts. This contention is not noticed by the client before transmission of the burst because there is no acknowledgment mechanism in OBS and, consequently, the optical burst is also lost.

There are many studies in the literature for reducing the burst loss rate. These techniques can be employed at the edges of the network or in the core routers. Edge-assisted contention avoidance techniques are less expensive than the techniques employed in the core of the network because they are easier to employ. One of the edge-assisted techniques is path switching which means alternating transmission paths to a destination depending on the congestion in

the network. This approach can reduce loss rates especially when some of the nodes in the network become highly congested.

In addition to networking techniques, it is possible to use some features of grid computing to reduce loss rate in the network. For example, consumers in a grid environment can choose from more than one resource to execute the grid job and this flexibility can be used to reduce loss rates by selecting the resources which have less congested incoming links. In this thesis, we develop a resource switching algorithm in combination with path switching to take advantage of this flexibility.

In addition to reducing loss rate, there are studies for supporting different service classes in an OBS network. One of them is to apply an extra QoS offset to high priority bursts and it is shown to reduce loss rates of high priority bursts. We assume that there are two different classes in the grid network and grid bursts constitute the higher priority traffic. In this thesis, we analyze the effect of using an extra offset for high priority grid bursts. Although increasing the offset times reduces the burst losses, it also increases the grid job completion times due to additional delay. We develop an algorithm for computing a QoS offset which minimizes the completion time of grid jobs by finding a balance between reduction in loss rate and increase in delay.

We first investigate the OBS grid architectures in the literature and offer improvements to these architectures for feedback collection and loss notification. These modifications are necessary to perform congestion-based decisions and to perform reliable transmission using the OBS protocol. These changes requires minimal signaling support and mostly integrated to the OBS grid architecture.

Next, we propose a congestion-based joint path and resource switching algorithm which is used to reduce loss rates in an OBS grid network. Since the consumers in a grid can request service from several providers, it is possible

to select resources and paths considering the congestion in the network. The proposed joint resource and path selection algorithm exploits the possibility of selecting from a number of resources in a grid computing environment. It is possible to utilize less congested parts of the network and distribute the traffic evenly using this method. This mechanism outperforms algorithms which perform resource selection and path selection separately especially for high levels of traffic load and it is the first edge-assisted network-aware resource selection strategy for OBS grids.

This scheme is then extended using an adaptive offset based QoS algorithm which computes offset value for grid bursts minimizing the average completion time. Although applying an extra offset to grid bursts increases transmission delay, average completion time can be reduced by decreasing loss probability. Numerical results show that the proposed algorithm achieves smaller job completion times compared with using fixed offset especially under non-static traffic conditions. Both of these methods offer grid-specific improvements to the underlying burst loss problem of burst switching and reduces burst loss rate.

In Chapter 2, we present a literature review of burst switching, grid OBS architectures and grid workload models. The algorithms proposed in this thesis are explained in Chapter 3. Chapter 4 and 5 present the numerical results and conclusions, respectively.

Chapter 2

Literature Review

This chapter presents a review of the literature to give background information related with this thesis. Since this study is based on improving quality of service for optical burst switched grid networks, this chapter includes information about both optical burst switching and grid computing. After a background information on OBS, contention resolution and avoidance techniques and service differentiation methods for OBS are explained. Separate sections are devoted to path switching and QoS offset based differentiation because these techniques are employed in this thesis. Next, OBS based grid architectures in the literature is presented. The chapter ends with the explanation of the parallel workload model we used in simulations.

2.1 Optical Burst Switching

Optical Burst Switching(OBS) is a switching paradigm which offers sub-wavelength granularity for optical networks [8]. In this mechanism, a control packet is sent before the optical data to configure switches on the path. After an offset time, the optical burst is sent without waiting for an acknowledgment

and OBS is a one-way reservation protocol because of lack of acknowledgment. In contrast to optical packet switching in which the header and the data are bonded, there is no need for buffering of optical data at each node for processing of the header because the optical burst and the header are separated in time domain. The optical burst is delayed at the source node waiting for the control packet to configure an all-optical path. The wavelength resources are released after the transmission of the burst automatically or after the reception of an explicit release packet. If a control packet fails to find an available wavelength, the optical burst is dropped at the node.

An OBS protocol called Just-Enough-Time(JET) proposed in [8]. In JET, the switches are not immediately configured when the control packet is received but the reservation is delayed until the expected reception time of the burst. This reduces bandwidth waste because another reservation can be made before the reception of the burst. Figure 2.1 illustrates the JET protocol. When the source node aggregates a burst, it first sends a control packet to the destination using a dedicated wavelength for signaling which is called control channel. When this control packet is received by the subsequent nodes, it is converted from optical domain to electrical domain and processed. The switches are reserved for the burst transmission duration and the control packet is forwarded to subsequent nodes after being converted to the optical domain again. The time required for opto-electronic conversion and processing is denoted as Δ. The optical burst is sent using the chosen wavelength after an offset time, T. The duration of the offset time is ΔH where H is the number of hops between source and destination.

The fundamental issues with OBS is to reduce burst drop rates and to handle burst contention [10]. Techniques employed at the edges of the network to reduce burst contention probability are called contention avoidance techniques and the techniques which are used by the routers to resolve contention are called

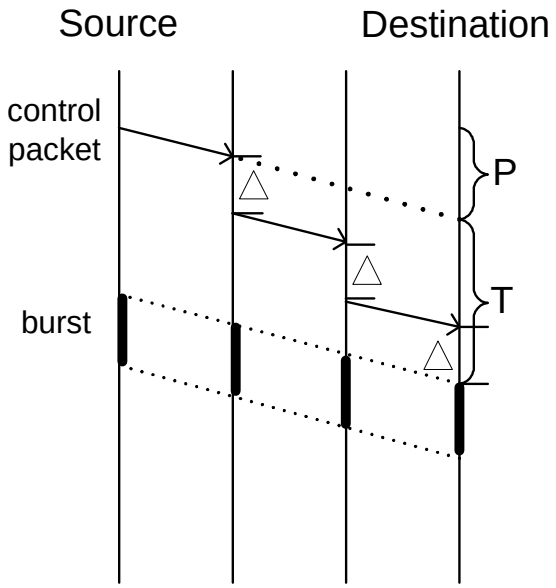

Figure 2.1: Timeline of the JET protocol.

contention resolution. In the next section, contention avoidance and resolution techniques will be explained briefly.

2.2 Contention Resolution and Avoidance

Since the wavelength reservation are made using one-way reservation, multiple bursts may arrive to a router contending for the same channel. In this case, one of bursts is dropped if there is no contention resolution technique is employed. It is possible to resolve contention using deflection in space, time and wavelength domains. Deflection in space domain is called deflection routing which is forwarding the burst through an alternative path. Deflection in wavelength domain is wavelength conversion which can be performed using wavelength converters and deflection in time domain means delaying a burst using fiber delay lines. If it is not possible to deflect a burst, there are also several techniques to reduce data loss in case of contention. Contention avoidance techniques are explained briefly as follows

- Wavelength conversion: When more than one burst contend for the same wavelength on an outgoing link, one of them can be switched to another wavelength channel if there is a wavelength converter. In full wavelength conversion, all wavelength channels can be switched to other wavelengths, however, in partial wavelength conversion there are limited number of converters so a limited number of channels can be switched at the same time. This is the most effective way of deflection but the technology is expensive and immature. Exact calculation of blocking probabilities in an OBS network with partial wavelength conversion is given in [11].

- Deflection routing: In this method, each OBS router knows two different paths to each destination. If it is not possible to reserve resources along the preferred path, the router forwards the control packet to the second path. This method does not require extra hardware but it may result in out-of-order packet delivery. Also, the required offset time is increased because of the possibility of using longer paths. Performance analysis of deflection routing for OBS can be found in [12].

- Fiber delay lines: It is possible to delay an optical burst for a fixed amount of time using fiber delay lines if the burst can find an available wavelength at the end of the delay period. However, fiber delay lines increase transmission delay and become very large in volume. Performance modeling of optical-burst switching with fiber delay lines is presented in [13].

If contention cannot be resolved using the deflection methods above, burst segmentation [14] can be used to reduce data loss caused by contention. In burst segmentation, the overlapping part of the contending burst is discarded and the remaining part of the burst is forwarded. This method reduces data loss but detecting segments in an optical burst and recovering the partial burst are challenging problems.

Deflection techniques reduce loss rates but they have some disadvantages in comparison to contention avoidance techniques. First of all, extra hardware and software are required at the ingress routers to perform deflection. Also, deflection techniques give suboptimal results because they can only deflect bursts locally. On the other hand, edge-assisted techniques can distribute the traffic load evenly by having a global view of the network. Also these techniques are easy to deploy and upgrade because all of the complexity is kept at the edges of the network. Some of the edge-assisted contention avoidance methods are summarized as follows

- Wavelength assignment: Careful assignment of wavelengths reduces burst loss rate significantly by preventing wavelength contention at the ingress routers. This method is effective especially when there is no or limited wavelength conversion and can reduce the number of converters required for a desired loss rate which decreases hardware costs. Wavelength assignment problem is extensively studied in the literature for all-optical network with wavelength division multiplexing [15]. Wavelength selection and routing is also effective on the burst blocking reduction gained by wavelength conversion [16].

- Edge Scheduling: This technique is based on the fact that burst loss occurs when number of simultaneous burst arrivals exceeds number of wavelengths. For that reason, scheduling transmission of bursts at the edge of the network appropriately may reduce simultaneous burst arrivals at the ingress nodes so loss rates can be reduced. [17] proposes an algorithm with is based on delaying the optical bursts at the edges of the network beyond their required offset times which is shown to reduce loss rates significantly.

- Rate Control: Similar to the previous approach, the rate of traffic injection can be controlled at the edges to reduce congestion in the network. In [18], a TCP congestion mechanism is proposed for OBS networks.

- Traffic engineering: Optimization strategies can be used to distribute traffic in the network evenly using traffic load estimations. It is possible to reduce burst loss rates significantly but traffic load estimations have to be known previously. [19] presents an integer linear programming model and heuristics for traffic engineering in OBS networks to solve this problem.

- Dynamic path selection: Dynamic switching of pre-determined transmission paths between the source and destination reduces loss rates in the OBS network. This method uses feedback collected from core routers in order to compare and select from different paths between the source and destination. A variation of this method is to re-compute the routes to destinations periodically using congestion information as the distance metric. Path switching for OBS is first proposed by [20] and further investigated by [21]. These methods are explained in more detail in Section 2.2.1.

Since we applied path switching method to grid OBS networks in this thesis, it explained in detail in the following section.

2.2.1 Path Switching

Path switching for Internet traffic is proposed in [22]. This study takes advantage of the diverse path availability in the Internet to reduce packet loss. Since performance of Internet paths fluctuate, this method can increase quality of service especially for real-time multimedia applications [23].

Congestion based path switching for OBS networks is first proposed by [20]. In this paper, the authors propose a dynamic route selection technique using alternate fixed shortest paths and they also propose a dynamic route calculation technique.

In the first technique, the source knows two paths to a destination and the core routers send their congestion information to edge routers in the network. The congestion information is a binary signal which is set to one when the load level of a link exceeds a threshold value. The source sums congestion value of the links over the preferred and the secondary path and selects the one with less congestion. In the case of equality, it selects the preferred path.

In the second method, the source computes path to the destination periodically. When computing this path, the source can set the distance metric equal to the offered load of links or to a combination of congestion with distance or hop count. It is shown that dynamic route calculation and dynamic route selection from static routes reduce loss rate.

More adaptive switching methods are proposed in [21]. We compared the algorithms we propose with these algorithms so a summary of each algorithm is given below.

- **Weighted Bottleneck Link Utilization Strategy (WBLU)**: In this strategy, the most utilized link over a path is used in path selection in combination with the hop count. The utilization of link l at time t, $U(l,t)$, is defined as

  $$U(l,t) = \frac{\sum_{i \in Succ(l,t)} T_i}{Wt}$$

  where W is the number of wavelengths and $Succ(l,t)$ is the set of bursts that successfully transmitted until time t. Then, at time t the source routes bursts along the path $\pi_{z^*(t)}$ whose index is obtained using

  $$z^*(t) = \arg \max_{1 \le z \le m} \frac{1 - \max_{1 \le k \le |\pi_z|} U(l_k,t)}{|\pi_z|}$$

  where $\pi_z, z = 1, \ldots, m$ is the set of m candidate paths from source to destination and $l_k, k = 1, \ldots, |\pi_z|$ is the set of links comprising path π_z.

- **Weighted Link Congestion Strategy (WLCS)**: In this strategy, the source uses congestion information of all links over path to route the bursts.

The congestion level $c(l)$ of link l at time t is defined as the ratio of bursts that have been dropped at the link. Then, assuming link independence, the loss probability of candidate path π_z is given by

$$b(\pi_z, t) = 1 - \prod_{1 \leq i \leq |\pi_z|} (1 - c(l_i, t))$$

Then, the index of the path to route bursts can be found using the following metric:

$$z^*(t) = \arg \max_{1 \leq z \leq m} \frac{1 - b(\pi_z, t)}{|\pi_z|}$$

- **End-to-end Path Priority-Based Strategy (EPP):** Unlike previous strategies, this one uses burst loss rate for each path instead of individual links. This mechanism relies on feedback from the core routers about individual bursts. The priority of path is determined according to the following equation:

$$\text{prio}(\pi_z, t) = \begin{cases} 1.0 \\ \frac{\text{prio}(\pi_z, t-1) N_z(t-1)+1}{N_z(t-1)+1} \\ \frac{\text{prio}(\pi_z, t-1) N_z(t-1)}{N_z(t-1)+1} \end{cases}$$

$N_z(t)$ is also updated as $N_z(t) = N_z(t-1) + 1$ each time a new burst is transmitted on path π_z. At time t, bursts are routed through the path $\pi_{z^*}(t)$ whose index is obtained using the following metric:

$$z^* = \begin{cases} z, & \text{prio}(\pi_z, t) - \text{prio}(\pi_x, t) > \nabla \forall x \neq z \\ \arg\max_{1 \leq z \leq m} \frac{\text{prio}(\pi_z, t)}{|\pi_z|}, & \text{otherwise} \end{cases}$$

In this algorithm, if the priority of a path is greater than all other paths above a certain threshold, that path is selected independent of its hop count. However, if a single path is not better than other paths beyond a certain threshold in terms of priority, hop count of the path is also taken into account.

This study also proposes hybrid path selection algorithms based on machine learning techniques to change the path switching technique over time. However, in this thesis, we are interested in individual path switching strategies.

2.3 Service Differentiation for OBS

Service differentiation mechanisms are needed for different data flows carried over the OBS network and these methods has to be different from the methods for electronic networks since there is no optical buffer for optical networks. Lack of buffering prevents queueing at the ingress nodes, consequently, the use of many queuing based QoS techniques. Despite, there are several proposals for QoS differentiation for OBS networks [24]. The simplest of them is the extra offset based QoS which explained in detail in the next section.

2.3.1 Extra Offset Based QoS

A simple QoS mechanism based on extended offsets is first proposed in [25] which analyzes a two class scheme. Assume that there are two classes of service in the OBS network named class 0 and class 1. Class 0 corresponds to best-effort service such as plain data transfer and class 1 corresponds to the high priority traffic such as real-time multimedia. An additional offset time is assigned to class 1 burst which gives a high priority for wavelength reservation.

Figure 2.2 shows how service differentiation is obtained using extra offset. It is assumed that the required offset time is negligible in comparison to extra offset for simplicity. Let t_{ai} and t_{si} denote the arrival and service-start time of class i request, respectively. Since there is no offset for class 0 burst, the arrival time and the service-start time will be equal if the channel is available. However, the service-start time of class 1 burst is delayed by the extra offset time, i.e. $t_{s1} = t_{a1} + t_{offset}$, if the reservation is successful.

The reservation process would be in First-Come-First-Served fashion, if there was no extra offset for both classes. However, if one of the classes have extra offset, the interaction becomes more complicated. Figure 2.2-a illustrates the

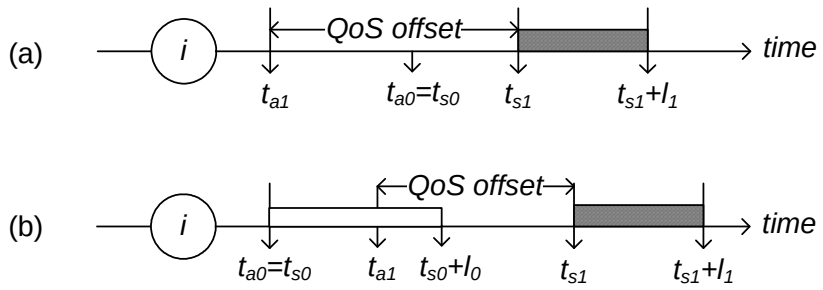

Figure 2.2: Illustration of extra offset time for obtaining service differentiation

situation when a class 1 request is received earlier than a class 0 request. Class 1 request is served but class 0 request is blocked if the $t_{s0} + l_L$ exceeds t_{s1}. In the second case, the class 1 request arrives later than the class 0 request. If there was no extra offset for class 1 burst, the request would be blocked since it arrived when class 0 burst is in service. However, if $t_{a1} + t_{offset} > t_{s0} + l_L$ the request is not blocked. As it can be deduced from this equation, blocking of a class 1 burst becomes independent from class 0 when t_{offset} is always greater than maximum burst length of class 0 bursts.

2.3.2 Upper and Lower Bounds of Blocking Probability

[25] also presents the lower and upper bounds of loss probability for both traffic classes. The lower bound for the class 1 occurs when the offset is larger than the maximum class 0 burst length, that is, when the high priority traffic is isolated from low priority traffic. In this case, the loss rate can be computed using Erlang's loss formula given in (2.1) assuming Poisson arrivals, i.e. $\pi_H = B(n, \rho_H)$ where n is the number of wavelengths and ρ_i is the traffic load of i^{th} class. The upper bound of class 1 blocking occurs when $t_{offset} = 0$ when all traffic behaves as if there are no classes. Then, blocking probability becomes $\pi_H = B(n, \rho)$.

$$B(n, \rho) = \frac{\frac{\rho^n}{n!}}{\sum_{i=0}^{n} \frac{\rho^i}{i!}} \tag{2.1}$$

If it is assumed that the overall blocking probability is independent that the number of classes and their offsets, which is called conservation law, it is trivial to find upper and lower bound on blocking probability of the low priority class using the following formula

$$\pi_L = \frac{\rho B(n,\rho) - \rho_H B(n,\rho_H)}{\rho_L} \tag{2.2}$$

The extension of this method for more than two classes of service is given in [26]. Also, performance of this scheme when FDLs are utilized is explained in [27]. In the following section, we present the performance evaluation of extra offset based QoS mechanism.

2.3.3 Performance analysis of QoS offset

Performance analysis of QoS offset for a two class system is given in [28]. This analysis also depends on the conservation law assumption.

The overall loss probability of OBS traffic can be computed using the Erlang B formula for an offered load A_O and n wavelengths.

$$\pi_O = B(n,\rho) \tag{2.3}$$

To find the loss probability of the high priority traffic, the affect of the low priority traffic on the high priority traffic must be considered. It is possible to write the loss probability of high priority traffic as

$$\pi_H = B(n, \rho_H + Y_L(\delta_H)) \tag{2.4}$$

where $Y_L(\delta_H)$ is the low priority traffic which is seen by the high priority traffic with a QoS offset of δ_H. Then, the loss probability of the low priority traffic can be approximated using the conservation law as

$$\rho_O \pi_O = \rho_H \pi_H + \rho_L \pi_L \tag{2.5}$$

where ρ_L is the offered load of low priority traffic. The low priority traffic affecting the high priority traffic, $Y_L(\delta_H)$, can be computed using

$$Y_L(\delta_H) = \rho_L(1-\pi_L)(1-F_L^f(\delta_H)) \tag{2.6}$$

where $\rho_L(1-\pi_L)$ is the low priority traffic which is not lost and $F_L^f(\delta_H)$ is the distribution function of residual life of low priority burst length. Since there is a mutual dependency between π_H and π_L, these equations has to be solved iteratively. The iteration begins with the upper bound of high priority traffic and lower bound of the low priority traffic as

$$\pi_H^{(0)} = B(n, \rho_H) \tag{2.7}$$

$$\pi_L^{(0)} = 1/\rho_L(\rho_O\pi_O - \rho_H\pi_H^{(0)}) \tag{2.8}$$

The the distribution function of the residual life of burst length is given by

$$F_L^f(t) = 1/h_L \int_{u=0}^{t} (1-F_L(u))du \tag{2.9}$$

where h_L and $F_L(u)$ represent the mean and the distribution function of burst transmission time, respectively. Using this formula, it is possible to calculate the low priority traffic seen by the high priority traffic as follows

$$Y_L^{(0)}(\delta_H) = \rho_L(1-\pi_L^{(0)})(1-F_L^f(\delta_H)) \tag{2.10}$$

After the initial values are computed using (2.7) and (2.8), the value found using (2.10) is inserted to (2.4) to for the second step in the iteration. After several iterations, the loss rate for each class can be obtained.

In the next section we explain the exact analytical modeling without conservation law assumption.

2.3.4 Exact modeling

An exact model for computing blocking probability for multi-class systems is presented for single wavelength in [29]. The authors extend their work for multichannel systems in [30]. This study uses concepts of burst contention window

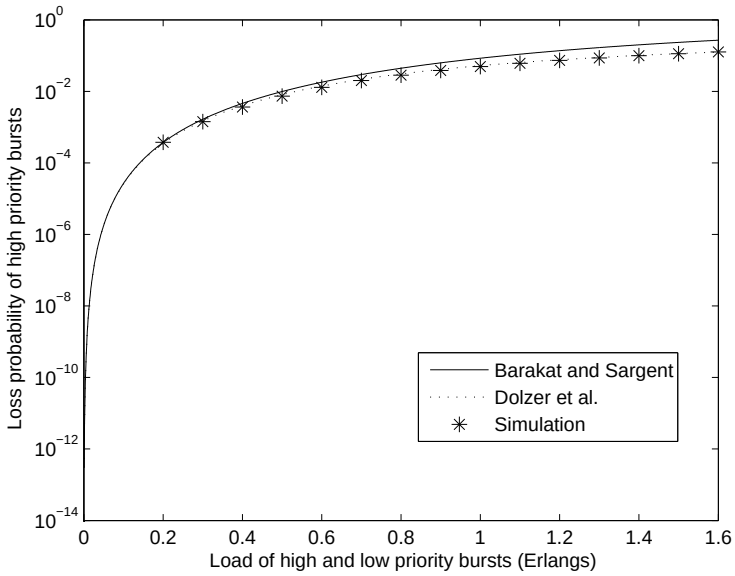

Figure 2.3: Graph of high priority traffic loss estimations and actual loss rate found by simulations.

and maximum perceived load to model the blocking probability. However, this formulation assumes blocking probability is very small than 1, $P_b \ll 1$. The scenarios that occur in OBS grid applications do not always satisfy this low loss probability assumption due to contentions. For that reason, we prefer to use the first model, although it assumes work conservation.

We compared these models for a specific situation to understand which one is better suited for our purpose. We analyzed the blocking probability for a single link for 2,000,000 bursts. The burst length distribution is uniformly distributed between 0.5 and 15 ms. The offset value for high priority bursts is 3 ms. The results are obtained for increasing traffic load which is chosen same for high priority and low priority traffic. The blocking probability estimations and actual blocking rate can be compared in Figure 2.3. This figure shows clearly that the difference between the actual loss rate and analytical loss estimation given by the model of Barakat and Sargent is increasing for higher loss levels.

2.3.5 Other Service Differentiation Mechanisms

The drawback of the extra offset based mechanism is the increased end-to-end delay. The loss rates of high priority bursts is reduced but the delay increase reduces the quality of service. For that reason, several other methods are proposed for service differentiation in OBS.

[31] proposes an intentional dropping scheme to maintain proportional QoS in the OBS network. This provides a controllable burst loss ratio for different service classes. Another edge-controlled scheme for QoS differentiation is presented in [32] which uses threshold-based prioritized burst assembly mechanism. The packets belonging to different service classes are assembled using different burst assemblers which have different threshold values. This threshold value affect the loss rates by generating bursts with different lengths and different delays satisfying QoS requirements.

In addition to these edge-controlled schemes, there are some other mechanisms which are implemented at the ingress routers. Prioritized contention resolution along with prioritized burst assembly is proposed in [33]. This scheme is based on selective segmentation, deflection and dropping is applied at the ingress router depending on the service class of the burst. [34] proposes and analyzes a preemptive wavelength reservation scheme. A low priority burst may be preempted before or during the transmission by a high priority burst to satisfy QoS requirements. Also, a control plane protocol for both congestion avoidance and differentiation service based on Available Bit Rate algorithm for ATM networks is proposed in [35].

In the next section, OBS based grid architectures in the literature is explained.

2.4 OBS Grid Architecture

Since OBS is a promising technology for grid applications as explained in Chapter 1, there are several proposals for an OBS based grid architecture. In this section, we explain these proposals in detail.

An architecture is first proposed by [36]. In this architecture, grid jobs are mapped into OBS bursts and information about the grid job is embedded to the burst header. The bursts are sent to the network without a specific destination address and they are deflected to suitable resources by the intelligent OBS routers. During this transmission both network resources and grid resources are reserved on the fly by intelligent routers. After an offset time, the optical burst carrying the grid job data is sent to the network. After the job is processed, the job result is sent in an optical burst with a specific destination address in contrast to the anycasted job burst because the destination of the job result is the user which sends the job.

In this architecture, the user is not interested in the selection of the grid resource as long as the requirements of the grid job is satisfied. Anycasting is used when a request can be executed by more than one servers. The paradigm of anycasting is proposed to be used for Internet traffic [37], [38] and several anycasting algorithms for grid OBS architecture is proposed and analyzed in [39].

A different architecture is proposed in [40, 41] in which active networking is used for job specification dissemination. In an active network, routers can perform computations on the packet contents and modify these packets [42]. These computations can be specified by users and can be customized for a specific application. Difference of this architecture from the previous one is that anycasting is not used and the intelligent (active) routers are sparsely placed. The job specification is sent to the nearest active router in form of an optical burst and it

is multicasted to the other active routers by this router. These active routers performs resource discovery using the specifications in the active burst and multicast this active burst to other active routers. Then, each active router send an acknowledgment (ACK) or a negative acknowledgment (NACK) burst to the consumer about the situation of the resources and they reserve the grid resource for a limited time if it is available. After receiving all ACK and NACK messages, the consumer selects a resource and transmits the job data using an OBS burst.

Another architecture is proposed in [43] which is similar to [36]. In this architecture, job specification is transmitted using the control plane instead of using active bursts and all routers in the network are intelligent routers. There are two reservation mechanisms presented in this paper: In implicit discovery and reservation, the control packet of grid bursts are anycasted to a suitable resource by intelligent routers reserving both the grid resources and network resources and the burst is sent to the network by the consumer without explicit acknowledgment. In the explicit discovery mechanism, job specification is disseminated using the control plane and the intelligent routers in the network return an acknowledgment to the consumer. Then, the consumer selects the resource and sends the job burst. In this mechanism, anycasting is not used.

Implicit reservation is a faster method to execute jobs, because the consumer does not wait for the acknowledgments from the network. However, in this method, grid resource selection is performed by routers and may be sub-optimal. Explicit reservation is a more consumer controlled architecture where the grid resource is explicitly selected by the consumer.

2.5 Grid resource model

In order to perform a realistic simulation of an OBS consumer grid, a workload model for grid jobs is required. This model is used to generate grid job parameters, to schedule jobs at the grid resources and to estimate execution times of jobs in our simulations.

There are several studies which analyze the characteristics of the workload for specific grids [44] but there is no consumer grid realized in practice currently. The difference between the consumer grid concept and the current grid practices is that the casual computing jobs of consumers are executed on grid resources in consumer grids. However, in current grid practices, relatively large jobs of e-science applications are executed.

The consumers use such an interactive remote computing facility only if it enables them to use some applications which they cannot use otherwise. To realize this, execution times of jobs has to be reduced significantly. For that reason, it is needed to parallelize these jobs in order to be executed on multiple processors to decrease execution times of jobs. In a grid environment, computational resources have multiple processors and parts of the submitted jobs can be executed in parallel on these multiple processors. However, depending on the characteristics of the job, the number of processors that will be used in execution may be fixed or variable.

Today, most of the computational jobs in a grid are moldable jobs [45]. For a moldable job, the total computational power needed is known, but the number of processors which will be used for execution can be determined by the executing resource. This flexibility allows resources to schedule jobs according to the cost metric they choose.

To model parallel jobs in our simulations, we used Downey's model [46]. This model is used to estimate speedup obtained parallel execution of grid jobs. Speedup can be defined as

$$S_p = \frac{T_1}{T_p} \tag{2.11}$$

where T_p is the parallel runtime of a job on p processors and T_1 is the sequential runtime of the same job. The speedup obtained by executing a job on multiple processors does not change linearly as the number of processors increase and this fact affects the scheduling decisions made by the resource. Downey's speedup model estimates the speedup of a job using its average parallelism, A, and its variance in parallelism, V. Average parallelism is the average of parallelism of the program throughout its execution and the variance in parallelism is the change of parallelism of the program over time. The variance in parallelism is defined as $V = \sigma(A-1)^2$ where σ is the coefficient of variance in parallelism. The speedup formula is given as

$$S(n) = \begin{cases} \frac{An}{A+\sigma(n-1)/2} & \sigma < 1,\ 1 \leq n \leq A \\ \frac{An}{\sigma(A-1/2)+n(1-\sigma/2)} & \sigma < 1,\ A \leq n \leq 2A-1 \\ A & \sigma < 1,\ n \geq 2A-1 \\ \frac{nA(\sigma+1)}{A+A\sigma-\sigma+n\sigma} & \sigma \geq 1,\ 1 \leq n \leq A+A\sigma-\sigma \\ A & \sigma \geq 1,\ n \geq A+A\sigma-\sigma \end{cases}$$

Using this speedup estimation, resource can estimate the execution time of a job and schedule submitted jobs over multiple processors. There are several scheduling strategies in [46]. In our simulations, we used a simple scheduling strategy which allocates a number of processors equal to the average parallelism of the job, A. If A processors are not available at time of the job request, the resource postpones the execution of this job until A processors become available.

In our simulations, the processing characteristics of jobs are determined by three parameters: Job instruction count in Million Instructions (MI), average parallelism and variance in parallelism. Also, resources are characterized with

the number of processors and the processing speed of each processor in terms of million instructions per second.

In next chapter, we present a joint resource and path selection algorithm for an OBS based grid architecture. We use path switching based contention avoidance and extra offset based service differentiation techniques explained in this chapter to develop this algorithm.

Chapter 3

JOINT RESOURCE AND PATH SELECTION

In this chapter, we present a contention avoidance and service differentiation mechanism which minimizes grid job completion time for OBS grids. The chapter begins by discussing the OBS grid architectures in the literature and explains the motivations behind the modifications that we make to these architectures. After a detailed explanation of the architecture we study, the completion time of a grid job is analyzed to establish a mathematical model for completion time minimization. Using this mathematical model, a strategy for joint path and resource selection is proposed which performs resource switching in addition to path switching. A service differentiation mechanism which exploits the trade-off between increased delay and reduced loss rates obtained by extra QoS offset is proposed next.

3.1 Dumb Networks vs. Intelligent Networks

There is a long lasting debate in the networking community about choosing between intelligent or dumb network architectures. An intelligent network consists of smart routing elements which provide various services for applications that are transmitted in the network. In contrast to this, in a dumb network, the routers are not aware of the data they are transmitting and intelligent decisions are made at the edges of the network. For example, Internet architecture is a dumb network where routers are just aware of the IP layers which provides basic routing. Higher level protocols such as TCP are implemented at the edge routers making intelligent decisions such as congestion control.

Critics of dumb network model claims that some applications require special treatments by routers. For example, real-time streaming applications could be given a higher priority at the routers to be transmitted effectively. On the other hand, the deployment of intelligent networks are more expensive and require more complex protocols. Also, some users does not need the special features of routers.

When the OBS architectures presented in Section 2.4 are examined, it can be seen that most of them uses an intelligent networking paradigm for OBS based grid computing. Some of them rely on active routing concepts and some of them rely on the on-the-fly route determination and resource discovery.

However, it will not be practical to deploy OBS routers with special features for grid computing. For that reason, in this thesis, an architecture with edge routing is examined. The architecture that we study is similar to the explicit reservation proposed in [43]. However, in our architecture, the routers in the network are not intelligent, instead, the routers adjacent to grid resources performs resource querying. Also, resource selection is performed by consumers only.

In the next section, the OBS architecture that we study is explained in detail.

3.2 Studied OBS Grid Architecture

The OBS grid architectures in the literature are explained in 2.4. In this section, we explain the architecture that we study in this thesis and explain the modifications we made to the architectures in the literature.

3.2.1 Job Specification Dissemination

In the OBS architectures given in Sec. 2.4, resource querying is performed in a distributed fashion in order to maintain scalability and interactivity in contrast to current grid practices [43]. In [36, 43] where all routers are capable of resource querying, job specification is disseminated using anycasting reserving both network resources and wavelength resources. In [40, 41], the specification is multicasted to sparse intelligent routers. Since we simulate a dumb network, there are sparse intelligent routers in the network we simulate and multicasting between these routers is used for job specification dissemination.

In [41], the job specification is sent as an active optical burst. However, in [43], it is sent over the control plane. In our architecture, we also use the control plane for grid signaling.

3.2.2 Grid Resource Reservation

When the intelligent routers receive the job specification, they query the resources. In [43, 41], the intelligent routers send an ACK or NACK to the consumer about the availability of resources. However, binary signaling is not sufficient for resource selection when there are more than one available resources.

For that reason, we studied an architecture where the intelligent routers send processing time estimations to the consumer and consumers use this information to perform resource selection. There are also other metrics that can be transferred to the consumer such as processing cost but we use completion time as the single metric for simplicity. The intelligent routers reserve the grid resources for a limited time in order to guarantee processing time offers as in [41].

3.2.3 Resource and Path Selection

In contrast to [43], the resource selection is solely performed by the consumer in this architecture not by the intelligent routers. Path selection is performed by consumers and core routers does not perform anycasting. A list of two link-disjoint paths between each consumer-grid resource pair is computed and one of these paths to a resource or consumer is adaptively chosen for sending a burst. These link-disjoint paths are computed using an edge-disjoint path pair algorithm [47].

3.2.4 Network Resource Reservation

Wavelength reservation is separated from grid resource reservation in [41]. In this architecture, the job burst is sent after resources are queried using active networking. However, in [43], wavelength reservation can be performed at the same time with the resource reservation in both implicit and explicit reservation. Since the resource is selected using intelligent routers, it is possible make wavelength reservations at the same time.

In this thesis, we study a consumer-controlled architecture in which both grid reservation and wavelength reservation is performed by the consumer. The

consumer chooses the resource and the route to that resource and the routers does not perform intelligent decisions.

3.2.5 Notification of Burst Losses

The time required for retransmission of a grid job when a burst is lost is very high if there is no explicit notification of burst losses because the consumer can notice a burst loss when the job result is not received until its expected arrival time. For that reason, we added an acknowledgment mechanism to enable early notification of losses. This acknowledgment is sent by the receiving party to the sender using the control plane.

3.2.6 Resource Acknowledgments

In a consumer grid architecture, consumer may perform resource selection using several metrics. These metrics may include processing time of the job, cost of processing, grid resource states and network resource states. When performing anycasting, the resource selection is performed by intelligent routers using the specifications of the customers. Otherwise, customers explicitly choose the resource satisfying their specifications.

Although all of these constraints are important in the decision of the consumer, we used job completion time as the single metric to simplify the problem. In our architecture, the acknowledgments sent by the intelligent resources include only the offered processing time of the job.

3.2.7 Feedback Collection and Congestion Measurement

The analytical model of QoS offset given in Sec. 2.3.3 uses the offered load values for high priority and low priority traffic to compute loss probability of each traffic class and the sender has to know these values to compute a QoS offset. For that reason, these values must be recorded by the core routers and must be transferred to the sender.

These recorded load values are transferred to the consumer using the acknowledgment messages sent by the intelligent routers. These messages are sent over two link-disjoint paths in order to collect congestion information of disjoint paths. The core routers on these disjoint paths write the offered load information for both classes in these acknowledgment messages. When the consumer receives these messages, it acquires the offered load information for all routers on both paths and use this information to send the job burst.

Unlike consumers, resources actively probe the network to receive congestion information. They send probe packets to the consumer over two link-disjoint paths just before the completion of the processing. The consumer send these packets back to the resources using the paths they are coming from. The core routers write the offered load information to these packets and resource acquires the offered load levels when it receives these packets. So, the resource use load information for path selection when the grid job is finished.

3.2.8 Lifetime of a grid job

We can summarize the OBS grid protocol that we study as follows: The consumer sends the job specification to the nearest intelligent router using a control packet and this specification is multicasted to the other intelligent routers using the control plane. When an intelligent router receives the job specification,

it queries the resources that it is responsible and creates an acknowledgment message containing the completion time offered by the resource. The acknowledgment message is sent through two disjoint paths to the customer over the control plane and the routers on this path record their congestion information on these packets. After the consumer receives all acknowledgment messages, it selects a resource and sends the grid job as an OBS burst using one of the two link-disjoint paths. When the resource receives the burst, it sends an acknowledgment message to the consumer using the control plane. At the same time, the selected resource starts processing the job and the other resources clear their reservations when their respective timeouts expire. Just before the completion of the grid job, the resource sends probe packets to the consumer over the two link-disjoint paths using the control plane. Consumer sends these packets back to the resource and on their way back, these packets record congestion information of the core routers. When the job is processed, the resource sends the job result in the form of an optical burst over the selected path.

The flowchart of the lifetime of a grid job can be seen in Fig. 3.1. In the next section, we analyze the phases of grid job completion and present the completion time optimization strategies from the consumer's point of view.

3.3 Consumer-Side Optimization

In this section, we describe how the grid consumer chooses the grid resource, path and offset to minimize the grid completion time.

3.3.1 Completion Time and Retransmission Cost

The timeline of a successfully completed OBS grid job can be seen in Figure 3.2. The components of the lifetime of an OBS grid job are the following:

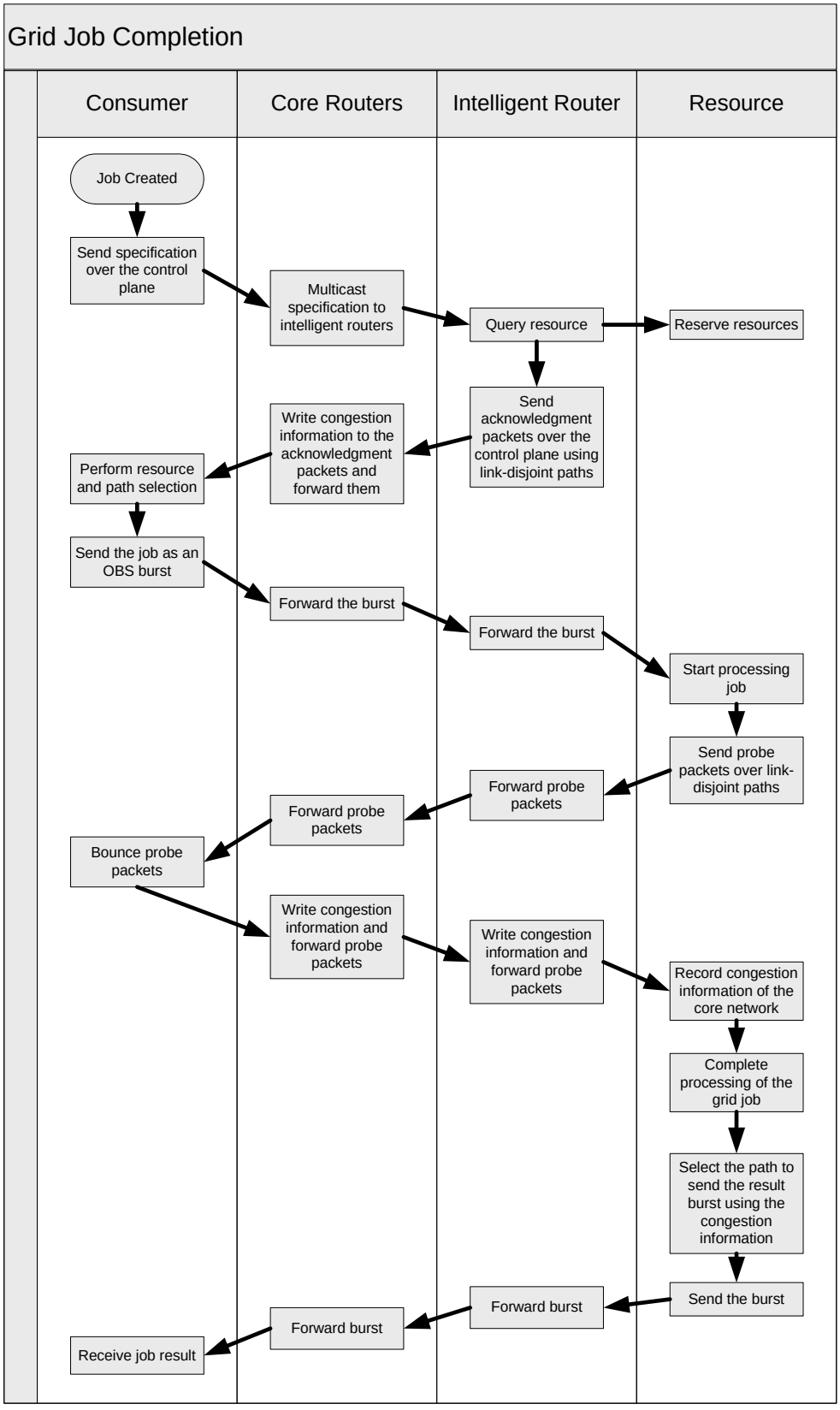

Figure 3.1: Flowchart of the lifetime of a grid job

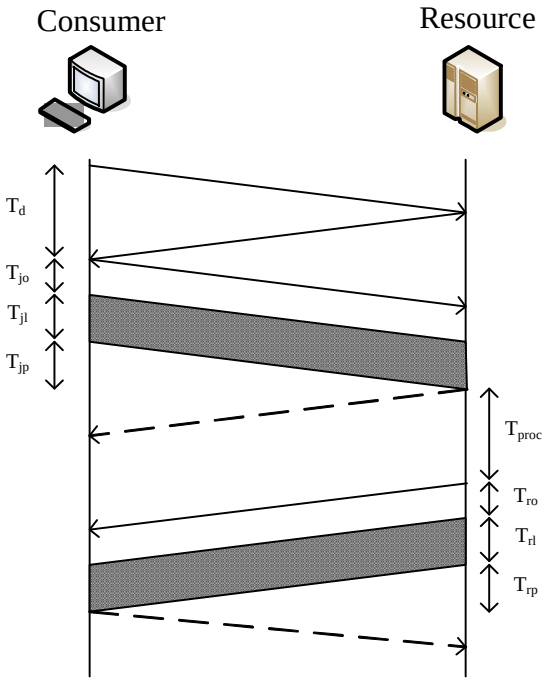

Figure 3.2: Timeline of a successfully transmitted grid job.

- T_d: Resource discovery delay
- T_{jo}: Offset time of the job burst
- T_{jl}: Transmission time of the job burst
- T_{jp}: Propagation delay of the job burst
- T_{proc}: Job processing time
- T_{ro}: Offset time of the job result burst
- T_{rl}: Transmission time of the job result burst
- T_{rp}: Propagation delay of the job result burst

From Fig. 3.2 it can be seen that the minimum required time to complete a job is

$$T_{min} = T_d + T_{jo} + T_{jl} + T_{jp} + T_{proc} + T_{ro} + T_{rl} + T_{rp}$$

For simplicity, we assume that the job result burst size is equal to the job burst size, i.e., $T_{jl} = T_{rl} = T_l$ and the propagation delay of the job burst is equal

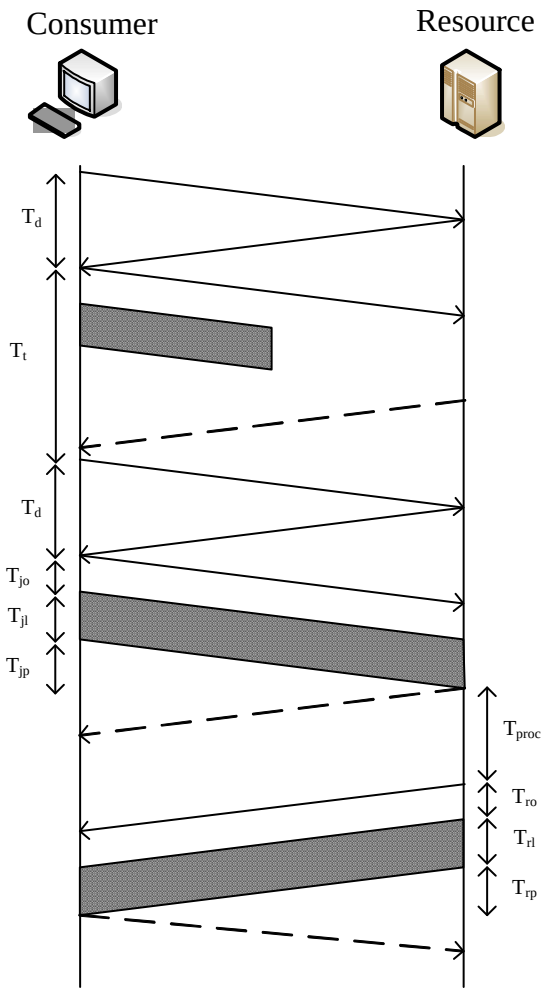

Figure 3.3: Timeline of a grid job when the job burst is lost.

to the propagation delay of the job result burst, i.e., $T_{jp} = T_{rp} + T_p$. We also assume that the required transmission offset, which is equal to the product of the number of hops on the path and the per-hop processing delay, is negligible with respect to other components. Under these assumptions the required time to transmit the job becomes

$$T_{min} = T_d + T_{jo} + 2T_l + 2T_p + T_{proc} + T_{ro} \tag{3.1}$$

However, if the job burst is lost, the time needed to complete the job increases. The timeline of a grid job when the job burst is lost once can be seen in Figure 3.3.

The consumer detects the loss after a timeout since it does not receive the burst acknowledgment sent by the resource. When a job burst is lost, time required to detect the loss of a burst is denoted as T_t. This timeout duration consists of job burst transmission delay, job burst propagation delay, job burst QoS offset and propagation delay of the burst acknowledgment. Timeout duration should also include a guard band, T_g, for unpredictable delays.

$$T_t = T_l + T_p + T_{jo} + T_p + T_g$$

In addition to the timeout duration, resource discovery phase has to be performed again because the computational resources reserve their processors for a limited time. Consequently, the retransmission cost, i.e., the difference between job completion time and T_{min}, is given by

$$\begin{aligned} T_{rt} &= T_t + T_d \\ &= T_l + 2T_p + T_{jo} + T_g + T_d \end{aligned} \tag{3.2}$$

Next, we use (3.1) and (3.2) to minimize the expected completion time of a grid job.

3.3.2 Expected Completion Time

Let $P_l^{(n)}$ be the loss probability of the grid job burst and $T_{rt}^{(n)}$ be the retransmission cost in the n^{th} transmission attempt, and, T_{min} is given by (3.1). Then the expected completion time can be written as

$$\overline{T} = T_{min} + \sum_{i=1}^{\infty}(\prod_{j=1}^{i} P_l^{(j)})T_{rt}^{(i)} \tag{3.3}$$

Assuming that the network and computational resource conditions does not change between transmission attempts, we have $P_l^{(n)} = P_l$ and $T_{rt}^{(n)} = T_{rt}$. Then,

the expected completion time of a grid job can be expressed as

$$\overline{T} = T_{min} + T_{rt}\frac{P_l}{1 - P_l}$$

This is the objective function which we want to minimize in this study. An algorithm which is not network-aware would make its choice using only the T_{proc} value which is the processing time of the grid job. However, in this study, we take into account the propagation delay between consumer and resource and the effect of blocking probability on the completion time. Next, we discuss how this expected retransmission cost can be used in resource and path selection.

3.3.3 Joint Resource and Path Selection

Each core router keeps a record of grid traffic and background traffic loads on its outgoing links. The length of bursts corresponding to each class is added to find T_{of}^{G} and T_{of}^{B} which are the total length of bursts offered to a link for grid traffic and background traffic, respectively. These values are set to zero periodically at the end of a predetermined time window in order to dynamically record traffic load changes over a link. The duration of this time window should be small enough to reflect short-term changes in the network and large enough to collect enough data about the traffic. At the end of a time window, the load on link l for each traffic class is computed using

$$A_l^G = \frac{T_{of}^{G}}{WT_{win}}, \quad A_l^B = \frac{T_{of}^{B}}{WT_{win}} \tag{3.4}$$

where T_{win} is the length of the time window and W is the number of wavelengths.

This load levels are transferred to the edge routers using acknowledgment and probe packets as described previously. Consumers can use this feedback to compute path loss probability of each disjoint path to a resource. When using this feedback from the core routers, consumer should also consider the traffic

generated by itself during the previous time window because most of the traffic load on a link might be generated by the consumer itself.

Let us denote the overall traffic load on link l which is received from the corresponding core router as A_l in Erlangs. Load level estimation when the traffic will be routed over this link can be expressed as

$$A'_l = A_l + \Delta$$

where Δ is the difference between the traffic offered by the consumer on link l between the next and previous time windows.

If the burst arrival distribution is Poisson, then the loss rate of link l when there are W wavelengths can be computed using the Erlang B formula given by Eg. 2.1.

Using the link independence assumption, the loss probability over path p can be written as

$$P_l^p = 1 - \prod_{l \in p}(1 - \pi_l)$$

For each resource-path pair, the consumer computes an expected completion time as follows.

$$\overline{T}^{r,p} = T_{min}^{r,p} + T_{rt}^{r,p} \frac{P_l^p}{1 - P_l^p} \tag{3.5}$$

where

$$T_{min}^{r,p} = T_d + 2T_l + 2T_p^{r,p} + T_{proc}^r$$

and

$$T_{rt} = T_t^{r,p} + T_d$$

assuming that no extra offset is used for job and job result bursts. After computing expected completion time for each resource and path pair, the consumer selects the pair (r,p) which minimizes $\overline{T}^{r,p}$, i.e., $(r,p) = \arg\min \overline{T}^{r,p}$.

Since all sources implement their own path switching algorithm independent of each other, grid traffic may oscillate if more than one source make similar

path switching decisions in a synchronized fashion. In this case, some near traffic sources may select paths which use same underutilized links at the same time making those links congested. After receiving load reports in the next time window, all of these sources switch away from those links in this case. These oscillations continue when all sources return to their first choices in the next time window causing higher burst loss rates. For that reason, we used a thresholding mechanism in order to prevent this kind of oscillations. In this mechanism, a source does not switch its resource and path choice in the previous time window unless more than 10% improvement obtained in estimated completion time.

In the next section, we present an adaptive offset based QoS mechanism for grid bursts which operates jointly with the resource-path selection mechanism.

3.3.4 Effect of extra offset for job bursts on completion time

Extra offset based QoS mechanism is used to guarantee a minimum burst loss rate for high priority bursts in the literature. However, the effect of the delay caused by the extra offset is application dependent and may be very significant for some time sensitive applications. For an OBS grid application, the extra offset can also be used to reduce the burst loss probability for high-priority grid bursts. However, the increase in the offset time will increase the minimum required completion time so the trade-off between delay increase and loss reduction needs to be addressed.

The minimum required completion time increases linearly in response to T_{jo} as it can be observed from (3.1). Similarly, the retransmission cost increases linearly with T_{jo}. However, it is possible to reduce the expected completion time function if loss probability can be reduced sufficiently.

In order to analyze the effect of offset time on completion time, we used the mathematical loss probability analysis explained in Sec. 2.3.3. In this model, there are two classes of traffic. We assume that grid bursts(job and result) constitute the high-priority traffic whereas all other bursts, called background, constitute the low priority traffic. Bursts belonging to both classes arrive according to Poisson processes. The overall loss probability of OBS traffic can be computed using the Erlang B formula for an offered load A_l and W wavelengths as given by (2.1).

To find the loss probability of the high priority traffic, the affect of the low priority traffic on the grid traffic must be considered. It is possible to write the loss probability of grid traffic as

$$\pi_l^G = B(A_l^G + Y_B(\delta_G), W) \tag{3.6}$$

where $Y_B(\delta_G)$ is the low priority background traffic which is seen by the grid traffic with a QoS offset of δ_G. Then, the loss probability of the background traffic can be approximated using the conservation law as

$$A_l \pi_l = A_l^G \pi_l^G + A_l^B \pi_l^B \tag{3.7}$$

where A_l^B is the offered load of the background traffic. The background traffic affecting the grid traffic, $Y_B(\delta_G)$, can be computed using

$$Y_B(\delta_G) = A_l^B (1 - \pi_l^B)(1 - F_B^f(\delta_G)) \tag{3.8}$$

where $A_l^B(1 - \pi_l^B)$ is the background traffic which is not lost and $F_B^f(\delta_G)$ is the distribution function of residual life of background burst length. Since there is a mutual dependency between π_l^G and π_l^B, these equations has to be solved iteratively as described in 2.3.3.

Effect of QoS offset on completion time

To understand the effect of offset on the completion time, we computed the average completion time with respect to traffic load and extra offset for a specific

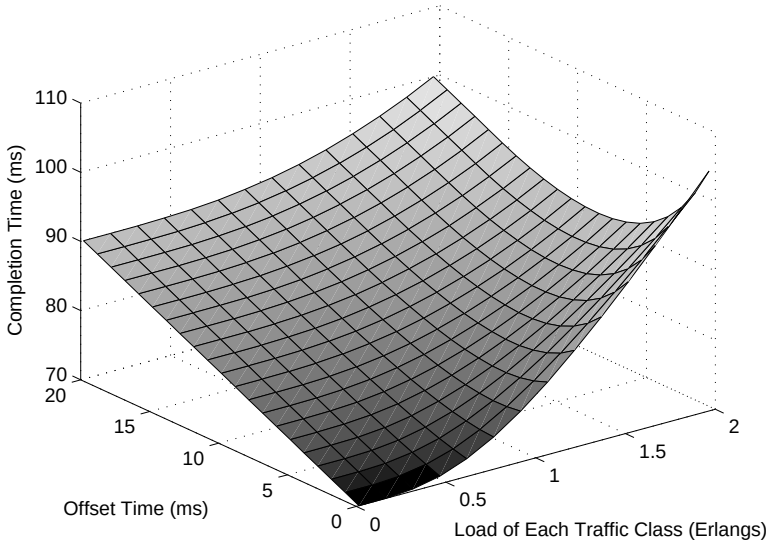

Figure 3.4: Completion Time vs. QoS Offset and Traffic Load

scenario. In this scenario, we assume a 3 hop path between the consumer and resource and 4 wavelengths at each link. Using the analytical model, we estimate the loss over the path with respect to different load levels and offset times, and used this loss value to compute the estimated completion time. The burst size distributions of each traffic class is between 0.5 and 15 ms. The change of estimated completion time with respect to offset and load can be seen in Fig. 3.4 for T_{min}=70 ms and T_{rt}=30 ms. From the figure, it can be deduced that applying an extra offset can reduce completion time for higher levels of traffic load. The graph of optimum offset which minimizes completion time with respect to grid and background traffic load is shown in Fig. 3.5. It can be understood that for very low background traffic optimum offset is 0.

3.3.5 Computing the optimum extra offset for a path

The feedback received from the core routers include the traffic loads generated by grid and background bursts. Using the same method explained in the previous

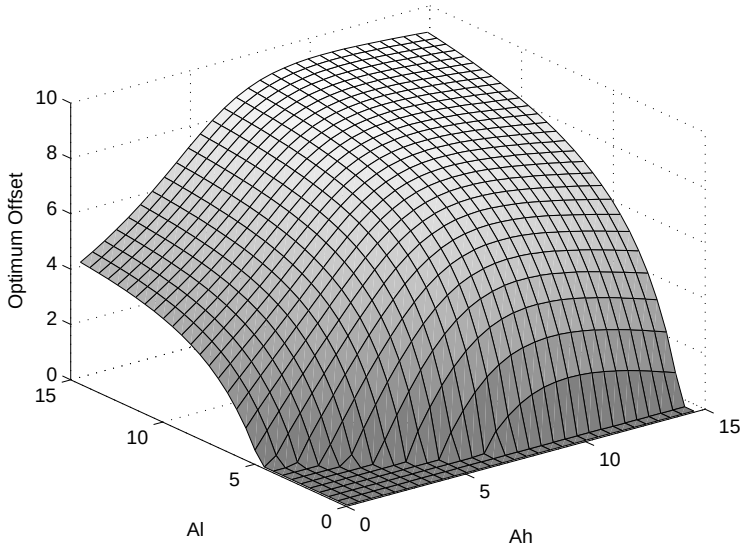

Figure 3.5: Optimum Offset vs. Traffic Load

section, the consumer separately estimates the grid load levels at each link on a path.

Then, the consumer performs a Fibonacci search to numerically find the value that minimizes the completion time function given in (3.5). Fibonacci search is a sequential single variable search technique to find the minimum of a function. Sequential search methods reduces the interval of uncertainty in which the optimum value can be found after performing several experiments. Fibonacci search uses Fibonacci numbers when determining the step size.

At each step of the Fibonacci search, the consumer computes the loss probability using the analytical model for the offset value and evaluates the completion time function using this value and offset. The interval in which the optimum completion time can be found is narrowed at each step and after a determined number of iterations, the optimum value is obtained with a certain sensitivity.

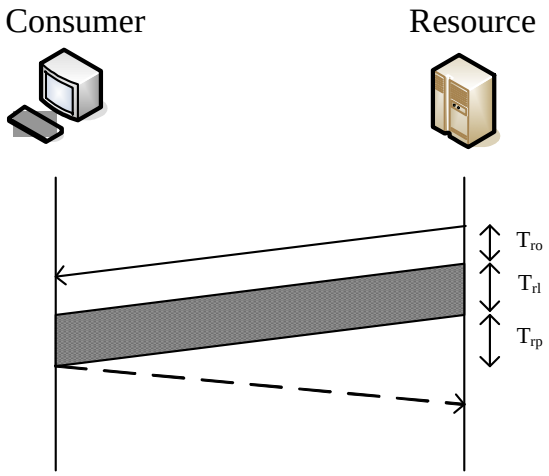

Figure 3.6: Timeline of a successfully transmitted grid job result.

3.4 Resource-Side Completion Time Optimization

In contrast to the consumer side optimization where the consumer can choose any resource to send the job, the only problem of the resource is to choose the path to send the result burst since the destination of the job result is readily known. Similar to the consumer, the resource also knows two disjoint shortest paths to consumer and it uses a similar approach to select the path to send the job result. The timeline of a job result burst which is successfully transmitted can be seen in Figure 3.6.

It can be seen that the minimum required transmission time for job result is

$$T_{min} = T_{ro} + T_{rl} + T_{rp}$$

If the job burst is lost, the retransmission cost is the timeout duration, which is required to notice the loss of the burst in addition to a guard band. The timeline of this situation can be seen in Figure 3.7.

$$T_{rt} = T_{ro} + T_{rl} + T_{rp} + T_g$$

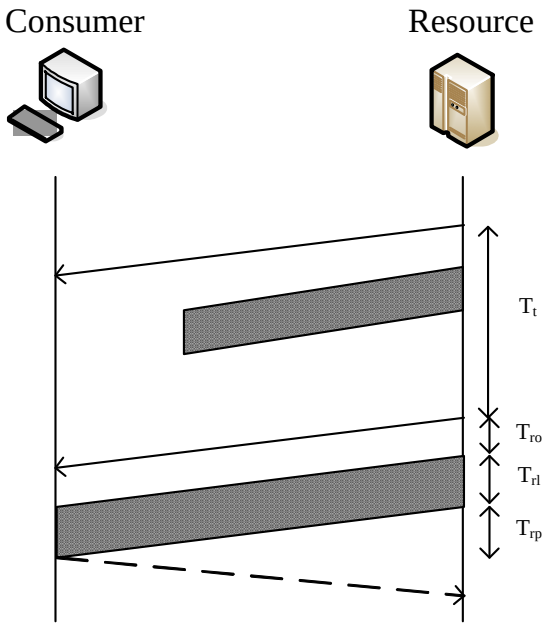

Figure 3.7: Timeline of a grid job when the job result burst is lost.

The difference of the extra offset mechanism for job result bursts from the one for job bursts is that the minimum required time and the retransmission cost functions changes. The retransmission cost of a job result burst is smaller than a job burst so it is expected that the optimum offset computed for job result bursts is smaller.

Chapter 4

SIMULATION AND RESULTS

In this chapter, we evaluate performance of the proposed joint resource and path selection algorithm in comparison to existing path switching algorithms under various scenarios. The path switching algorithms that we compare our algorithm and the simulation environment are explained in the beginning of the chapter. Then, the background traffic model that we have used in our simulations is defined. Evaluation of the proposed algorithm under stationary and non-stationary background traffic loads is given next.

4.1 Algorithms in Comparison

The proposed joint path and resource selection algorithm is compared with existing path switching algorithms. A resource selection method which selects the resource offering minimum computation time (MCR) is used in combination with these path selection algorithms. This method does not take the network congestion into account when selecting the computational resource but it selects the nearest resource if the resources offer the same computation time.

Existing path selection algorithms are as follows:

- Shortest Path Algorithm (SP)
- Weighted Link Congestion Strategy (WLCS): This path switching strategy is proposed in [21] and explained in Sec. 2.2.1. It computes the successful transmission probability of a path using the loss reports of each core router and weights this probability using the hop count of the path when selecting a path.
- Weighted Bottleneck Link Utilization Strategy (WBLU): This algorithm is also proposed in [21] and explained in Sec. 2.2.1. It uses the utilization value of the most congested link along a path weighted by the hop length and selects the path accordingly.

The algorithms proposed in this thesis are JR-NO, which is the joint resource and path selection algorithm with no offset, and JR-AO, which is the joint resource and path selection algorithm with adaptive offset. We also use a fixed offset mechanism JR-FO in which a fixed offset is applied to every burst in the grid network to evaluate the performance of adaptive offset selection.

4.2 Grid network model

The OBS grid network shown in Figure 4.1 is used in simulations where the length of each core link is indicated. In this topology, there are 7 customers and 3 resources. Each customer and resource is connected to the core network through an edge router. Also, there is an intelligent router adjacent to each resource which performs resource querying and sends acknowledgments to consumer regarding that resource.

The length of the background bursts and grid bursts is distributed uniformly between 0.5 ms and 15 ms. Each optical burst carries a single grid job or grid job result. We assume that the result of a job has the same data size with the

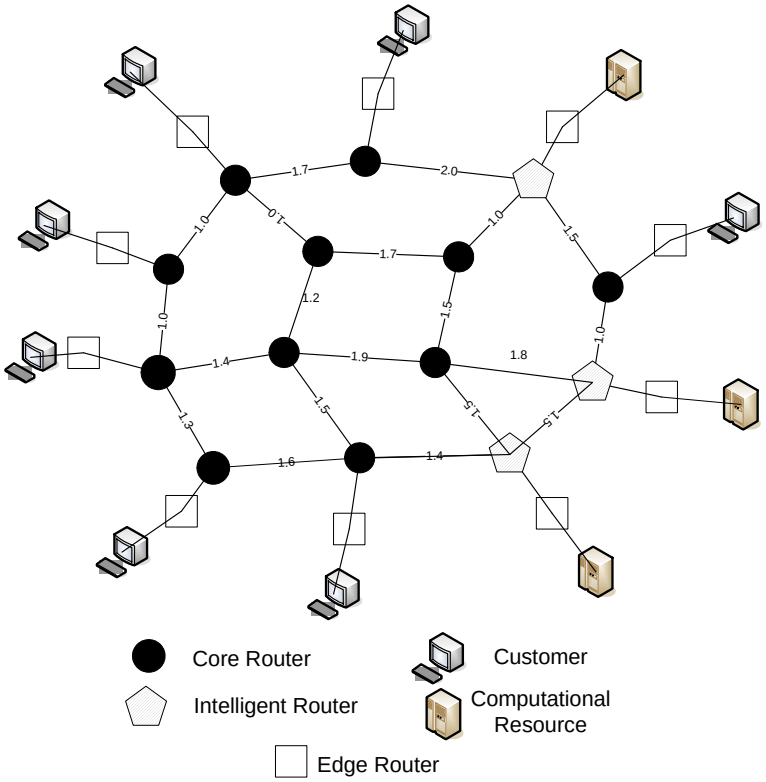

Figure 4.1: The simulated OBS grid topology. The numbers show the propagation delay of each link in ms.

job itself. The switching time for the core switches is 0.1 ms and control packet processing time is negligible. There are $W = 5$ wavelengths per fiber at each link and one of them is reserved for the control plane. Also, we assume that there are 10 links between edge routers and core network in order to prevent congestion at the edges of the network. The core routers record their load measurements periodically using $T_{win} = 1s$. Each simulation is performed for 300,000 jobs, however, the statistics of the last 50,000 is taken into account in order to ensure that the simulations reach a stable state.

As the resource model we used the Downey's model explained in Sec. 2.5. In this model, the processing characteristics of jobs are determined by three parameters: Job instruction count in Million Instructions (MI), average parallelism and variance in parallelism. We chose the job instruction count to be distributed uniformly between 100 and 3,000 MI and average parallelism distribution between 1 and 20. We take parallelism variance distribution between 0 and 2. Resources

are characterized with the number of processors and the processing speed of each processor in terms of million instructions per second. In simulations, each computational resource has 5000 processors and each processor has a processing power of 20,000 Million Instructions per Second (MIPS).

In order to evaluate the performance of the proposed congestion avoidance mechanism, we simulate a background burst traffic independent of the grid traffic. The background traffic model is explained in the following section.

4.3 Background Traffic Model

The characteristics of the background traffic which shares the network resources with the grid traffic have an important effect on the performance of path selection algorithm. If the background traffic is showing too little change over time, a path switching algorithm will not be necessary since the loss rates of alternative paths rarely change. On the other hand, if the distribution of the background traffic over the network is fluctuating, a path switching algorithm performs much better than a static routing algorithm.

We used an MMPP traffic model to emulate the background traffic because traffic load in communication networks are bursty. In the simulations, each edge router keeps an average of 3 flows at the same time to different edge routers and each flow has an average holding time of 120 seconds. Bursts belonging to these flows are generated according to an MMPP distribution. One of the states of the MMPP distribution is the high load state and the other one is the low load state.

The burst arrival rates at the states of an MMPP flow is determined according to a burstiness factor $\gamma \leq 1$. The traffic load is $L_h = L_{Av}/\gamma$ in the high load state and $L_l = L_{Av}\gamma$ in the low load state. We determine the average load per

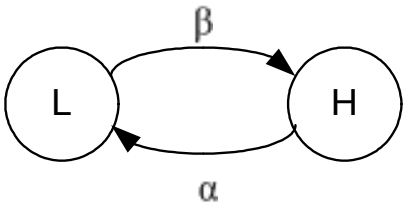

Figure 4.2: Markov chain representing the states of a Markov Modulated Poisson Process

flow, L_{Av}, to satisfy a desired offered load level on each link using the following formula:

$$L_{Av} = \frac{L * N_{links}}{S * F * N_{hops}} \tag{4.1}$$

where L is the desired average load level per link and N_{links} is the number of links in the network. S denotes the number of edge routers and F is the average number of background flows originating from an edge router at a time, we select $F = 3$ in our simulations. N_{hops} is the average number of hops that background bursts travel.

The transition rates of the MMPP distribution, α and β, are determined to satisfy the average load per flow, $L_{Av.}$. First, state probabilities are found by solving these two equations

$$L_l\pi_l + L_h\pi_h = L_{av} \tag{4.2}$$

$$\pi_l + \pi_h = 1 \tag{4.3}$$

Next, the transition rates can be found by selecting an appropriate value for one of the transition rates, α and β, and computing the other one using the formula

$$\pi_l = \frac{\alpha}{\alpha + \beta} \tag{4.4}$$

Using this model, it is possible to experiment with different burstiness levels by changing the value of γ. Traffic generated by each flow is static for $\gamma = 1.0$. The less the gamma is, the more bursty is the traffic.

4.4 Stationary Background Traffic Scenario

Dynamic path switching and adaptive offset schemes are expected to give better results under dynamic traffic loads because of their ability to react changes in the network. However, the proposed algorithms are first compared for a stationary background traffic load. In this case, "stationary" means that the average background load per link does not change over time. However, since each flow generates MMPP traffic, the traffic distribution is still bursty for $\gamma < 1$.

Simulations under stationary background traffic are performed for different values of background traffic load and burstiness factors.

4.4.1 Effect of Increasing Background Load

For $\gamma = 1.0$ and a grid load of 0.1 Erlang the graph of average completion time, job burst loss, result burst loss and average offset for changing background traffic load are given in Figs. 4.3, 4.4, 4.5 and 4.6, respectively.

In terms of completion time, it can be seen that JR algorithms perform better than MCR-WBLU and MCR-WLC which perform resource selection and path selection separately and also better than MCR-SP which uses shortest path routing. JR-AO performs better than JR-FO which is also better than JR-NO. Since JR-AO determines the offset value adaptively, it outperforms JR-FO which applies static offset to every burst in the network. JR-AO reduces completion time up to %5 in comparison to JR-FO and %10 in comparison to MCR-WBLU. All of these algorithms show similar performance for low background load levels but their performance differences become more visible for higher background loads. For that reason, it can be deduced that the resource and path selection algorithm is not crucial for low loads.

Fig. 4.6 shows the average offset for job and result bursts for different background loads generated by JR-AO. We selected the average of these offset values as the fixed offset value for all bursts in JR-FO. As it can be seen from this graph, the average offset value applied to the job bursts are larger than the average offset of result bursts because retransmission costs for job bursts is larger than the result bursts. For that reason, the average fixed offset value is generally larger than the offset value of result bursts and smaller than the offset value of job bursts.

Because of this difference, the loss rate graphs of job bursts and result bursts show different behavior. Loss rate of job bursts is significantly lower in JR-AO than JR-FO but loss rate of result bursts is slightly lower when JR-FO is used. It is important to note that JR-AO selects offset in order to optimize the completion time but not to reduce loss rates. For that reason, higher result burst loss rates is not a disadvantage for JR-AO.

In Figure 4.4, it can be observed that JR-AO and JR-FO perform worse than JR-NO in terms of job loss rate when background load is 0.2 Erlangs. However, the effect of increased loss rate is not significant in terms of completion time. The reason of this lower performance is the thresholding mechanism that we use to prevent oscillations. The initial job burst path choices in JR-AO and JR-FO simulations is different from the ones in JR-NO and these paths turn out to be worse. However, these algorithms does not change these initial choices although there are better alternatives since switching does not bring an advantage more than 10% in terms of completion time. This thresholding mechanism results in increased job loss rate but does not cause a significant worsening in completion time. Such a situation can occur in very low loads where switching choice does not have an important effect on completion time.

Performance graphics for more bursty traffic, for $\gamma = 0.25$, can be found in Figs. 4.7, 4.8, 4.9 and 4.10, respectively. It can be deduced that JR-AO algorithm

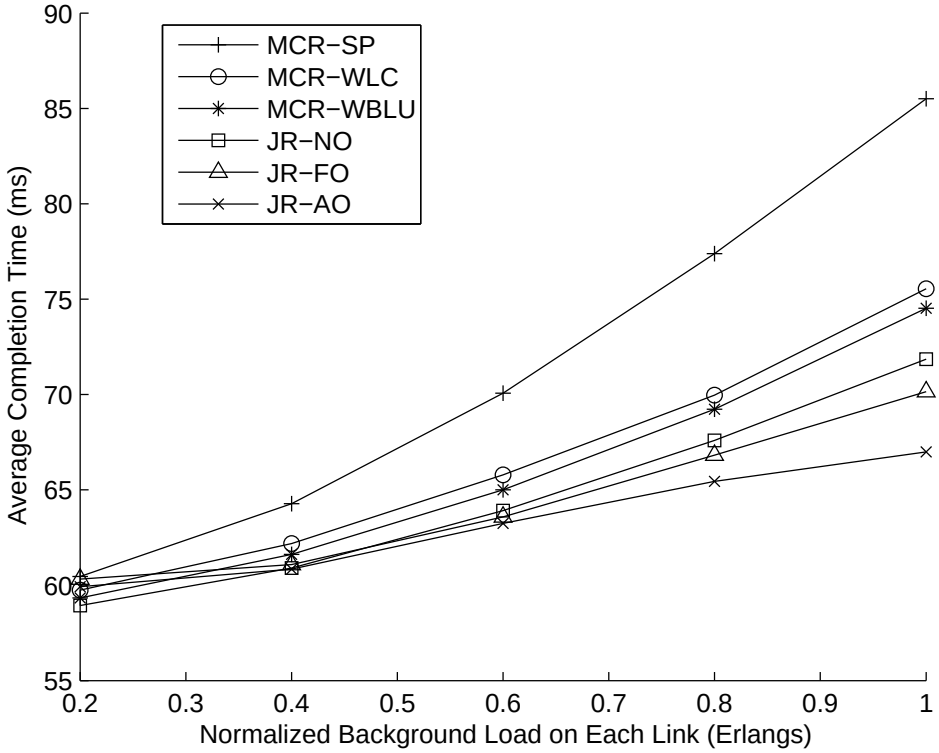

Figure 4.3: Graph of average completion time vs. offered background load for $\gamma = 1.0$

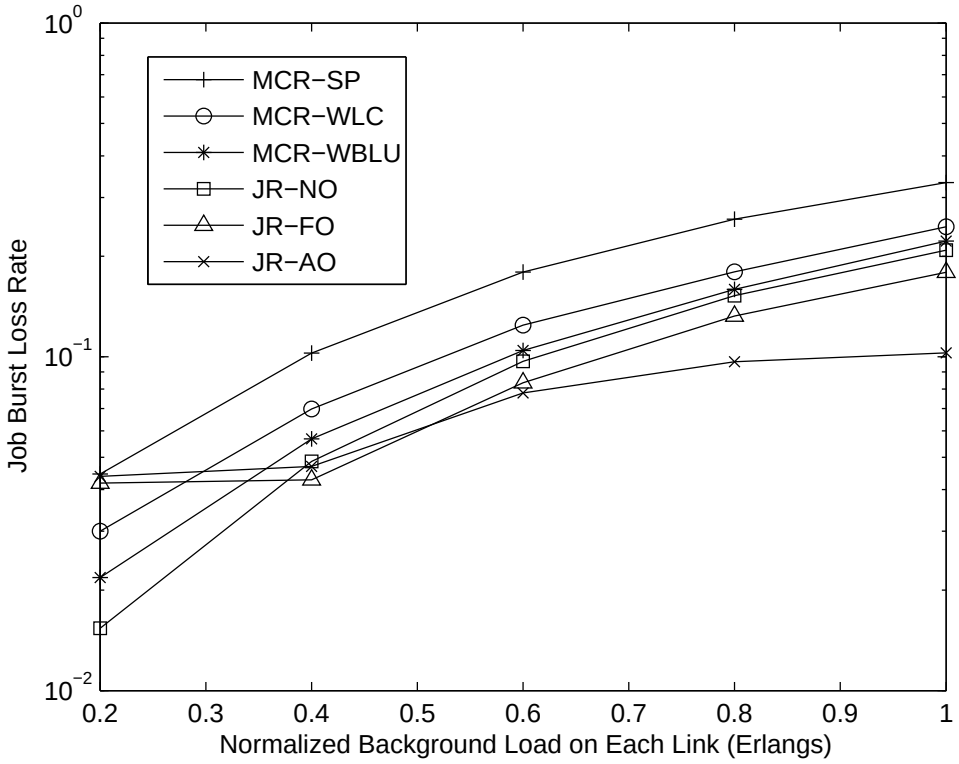

Figure 4.4: Graph of job burst loss rate vs. offered background load for $\gamma = 1.0$

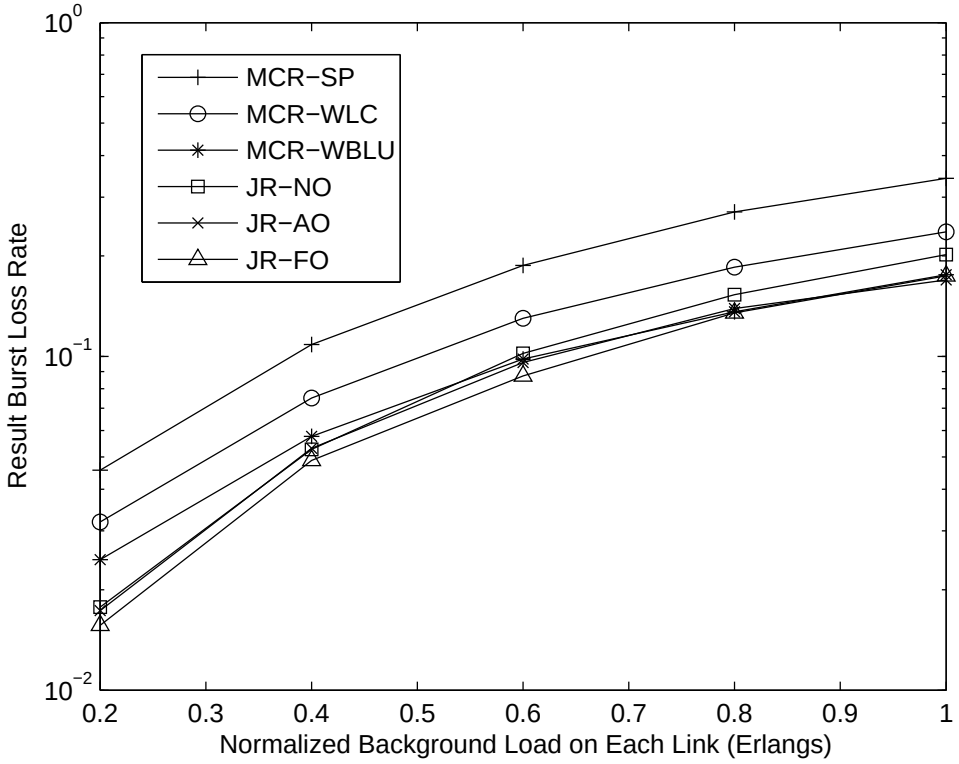

Figure 4.5: Graph of result loss rate vs. offered background load for $\gamma = 1.0$

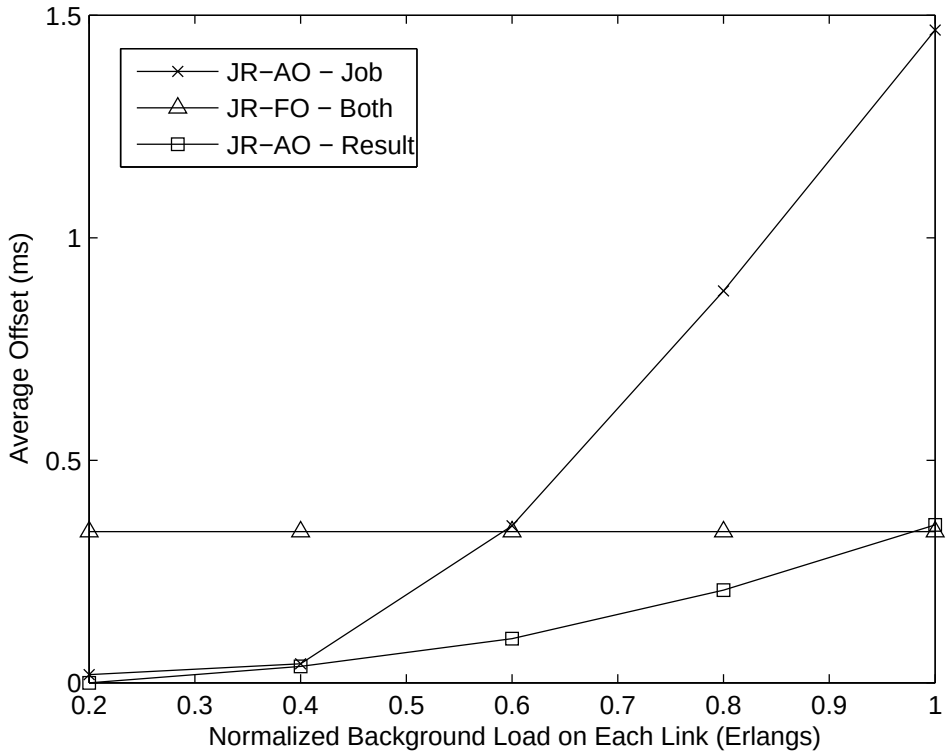

Figure 4.6: Graph of average extra offset vs. offered background load for $\gamma = 1.0$

makes a larger improvement with respect to other algorithms in this bursty traffic model and the performance difference can also be seen for lower loads. One of the differences between bursty traffic scenario and the stationary traffic scenario is that the MCR-WBLU algorithm perform better than the JR-NO algorithm which is explained in the next section.

4.4.2 Effect of Increasing Burstiness

As the burstiness of the background traffic load increases, estimation of loss rates become more difficult. We performed several simulations with different burstiness factors without changing the background load to evaluate the effect of burstiness factor, γ, on the performance of compared algorithms. Figs. 4.11, 4.12, 4.13 and 4.14 show the average completion time, job loss rate, result burst loss rate and average offset graphics, respectively, for different burstiness levels

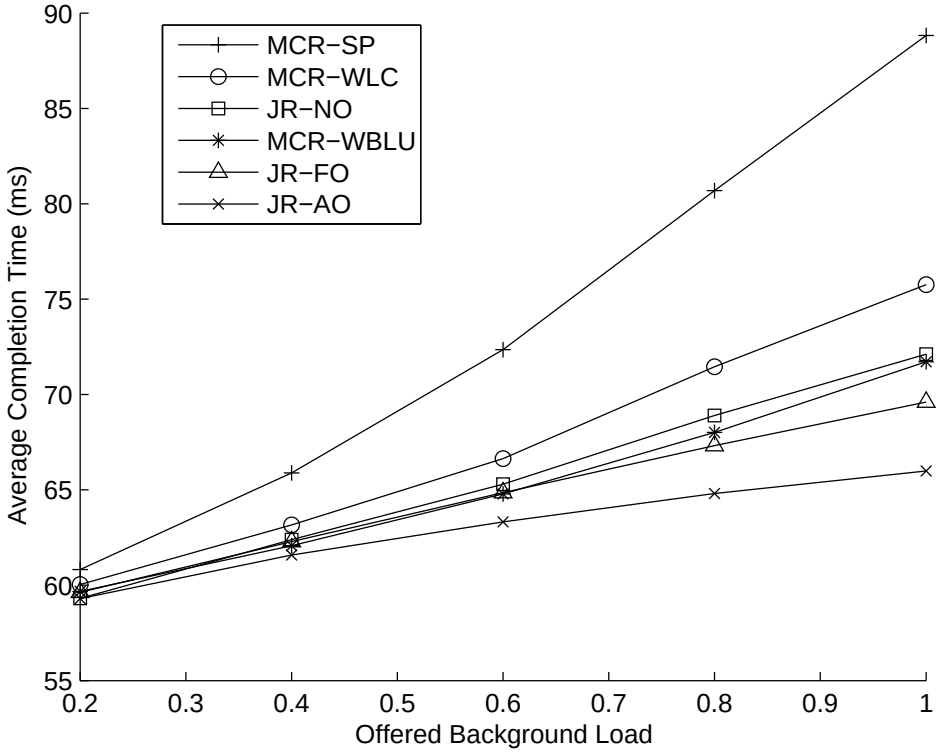

Figure 4.7: Graph of average completion time vs. offered background load for $\gamma = 0.25$

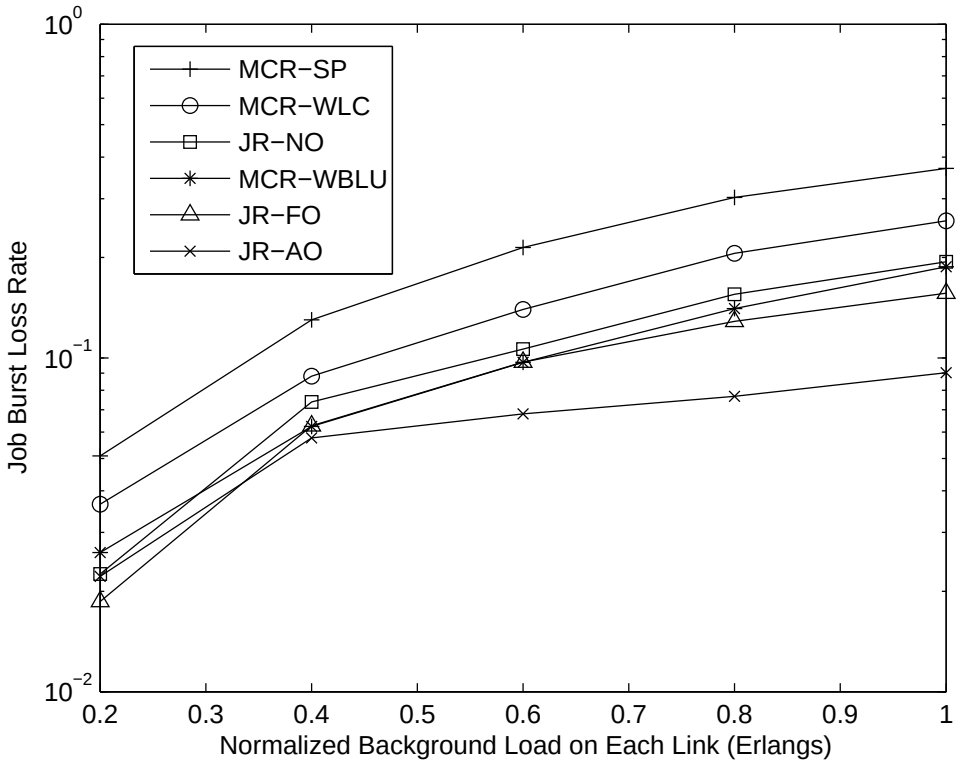

Figure 4.8: Graph of job burst loss rate vs. offered background load for $\gamma = 0.25$

when background load per link is 0.4 Erlangs and a grid traffic load per link is 0.1 Erlangs.

In terms of completion time, JR-AO performs best for all burstiness levels. As burstiness increases, average completion time for all algorithms increase except MCR-WBLU. MCR-WBLU starts to perform better because MCR-WBLU uses the load level of the most congested link over a path in path selection. As the burstiness increase, load differences between individual links become more significant so using only most congested link in path selection start to perform better.

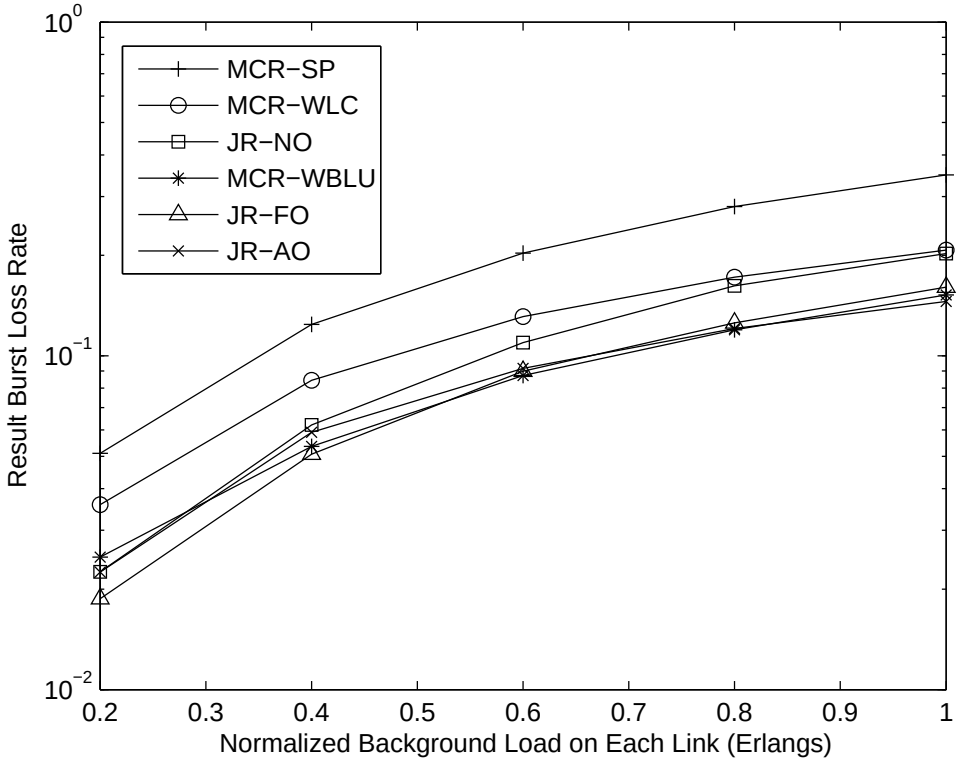

Figure 4.9: Graph of result burst loss rate vs. offered background load for $\gamma = 0.25$

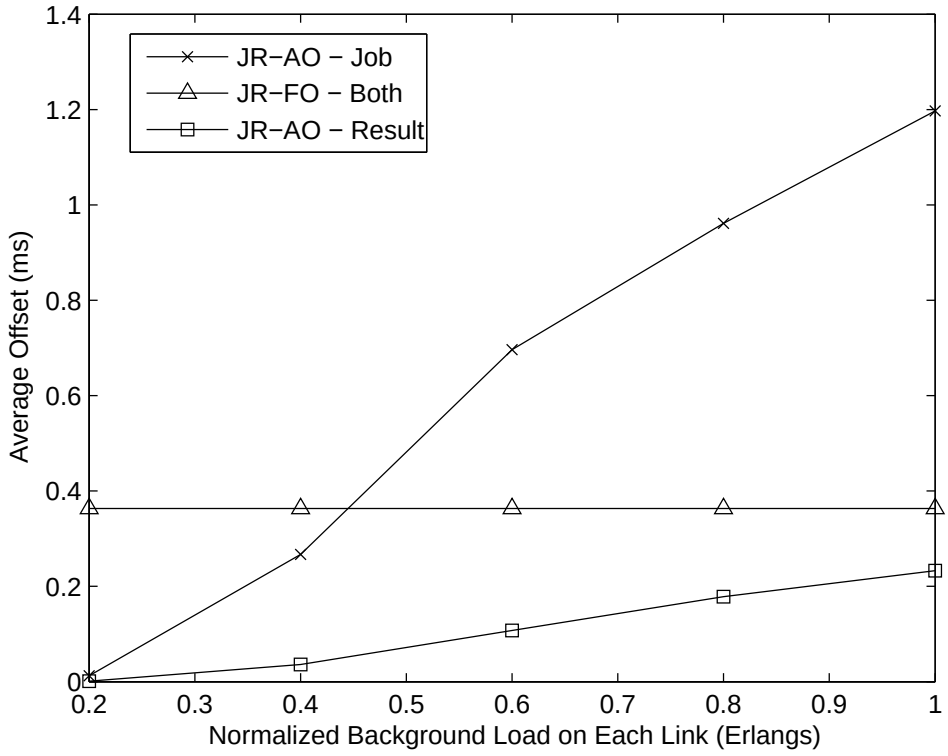

Figure 4.10: Graph of average extra offset vs. offered background load for $\gamma = 0.25$

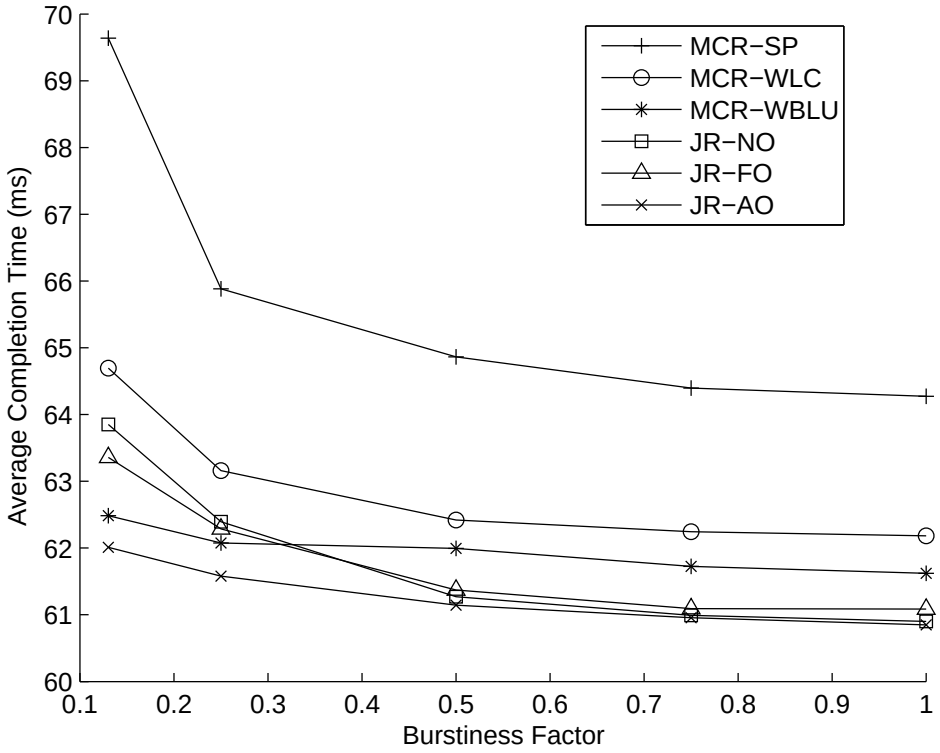

Figure 4.11: Graph of average completion time vs. burstiness factor γ

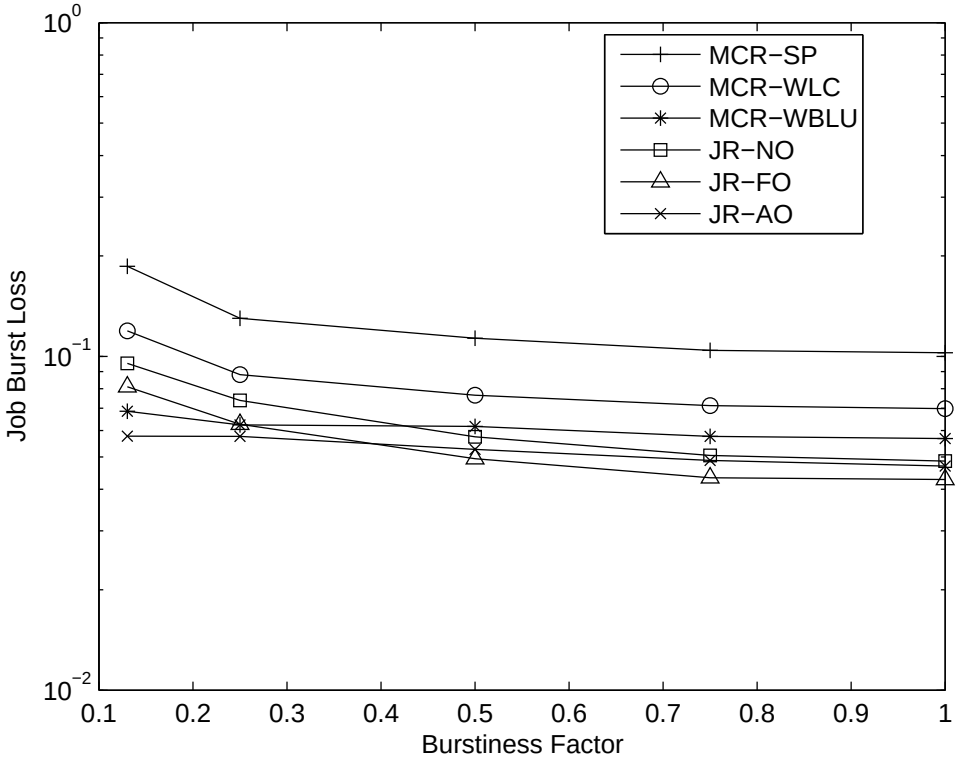

Figure 4.12: Graph of job burst loss rate vs. burstiness factor γ

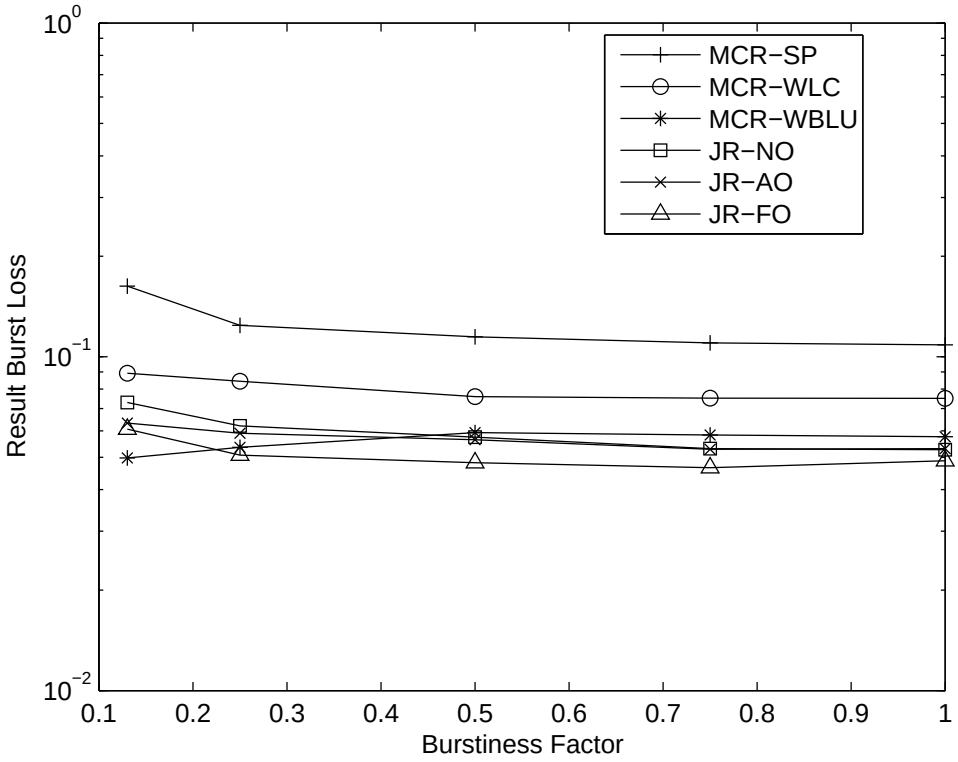

Figure 4.13: Graph of result burst loss rate vs. burstiness factor γ

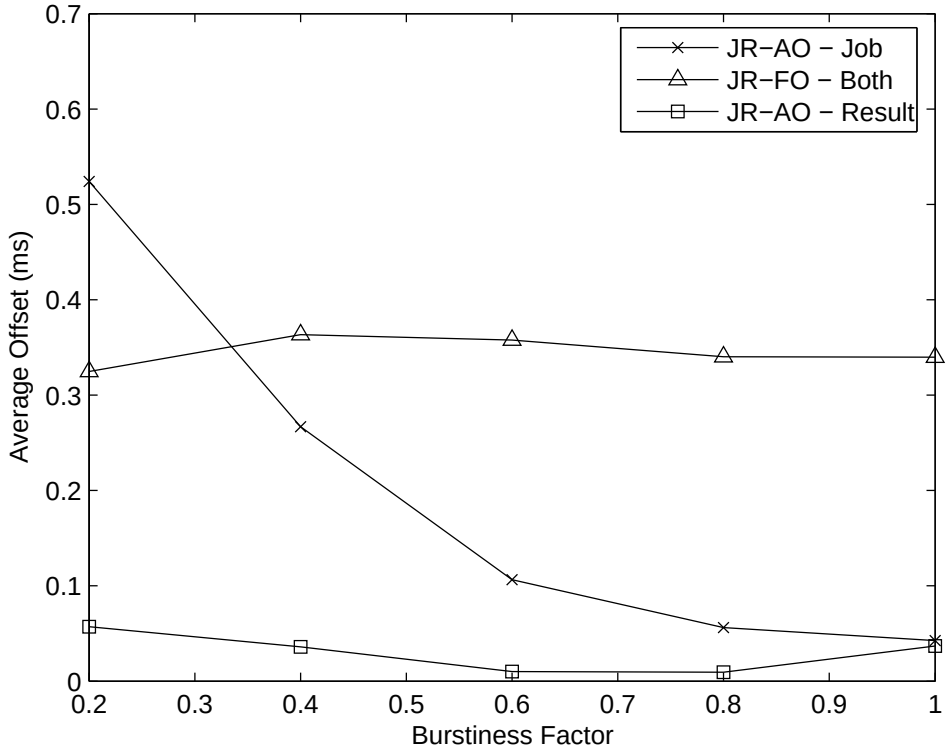

Figure 4.14: Graph of average extra offset vs. burstiness factor γ

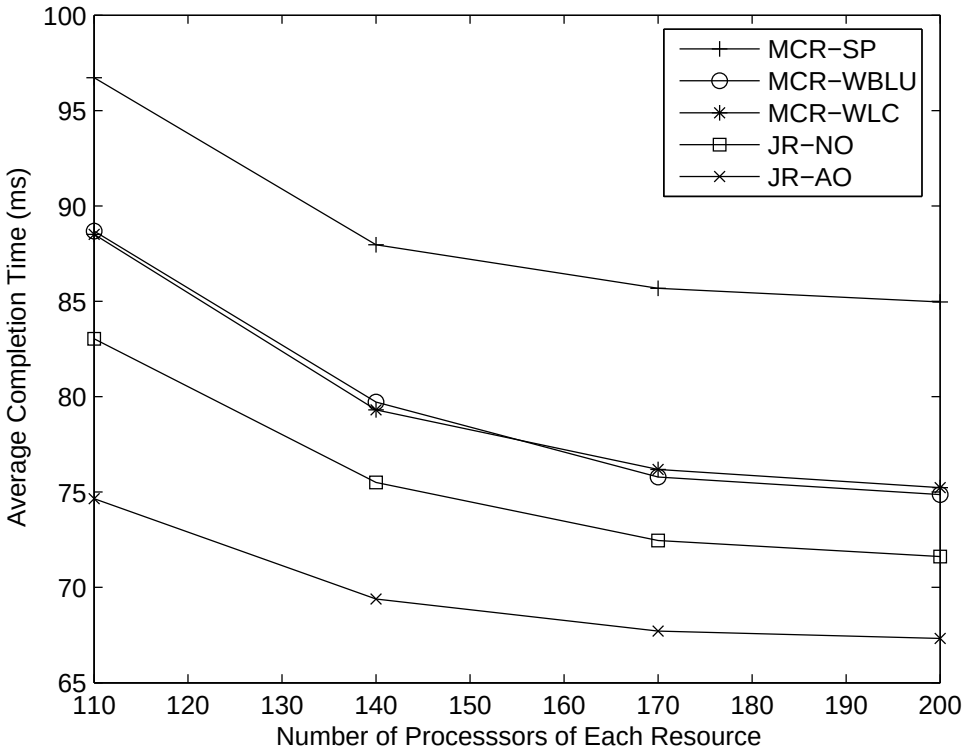

Figure 4.15: Graph of average completion time vs. number of processor for each resource

4.4.3 Effect of Resource Parameters

In addition to parameters related to network infrastructure, it is insightful to investigate the effect of change of grid job parameters on average job completion time. Although these parameters do not have an effect on the network statistics such as burst loss, they change average completion time because of the change in processing times. Figure 4.15 shows the change of average completion time versus the number of processors at each resource. Since the jobs are queued at computational resources, the decrease in the number of processors results in increased completion times. Further reduction causes infinite waiting times at the resources.

Figure 4.16 shows the change of average completion time for increasing mean of average parallelism distribution, A. A is distributed uniformly between 1 and

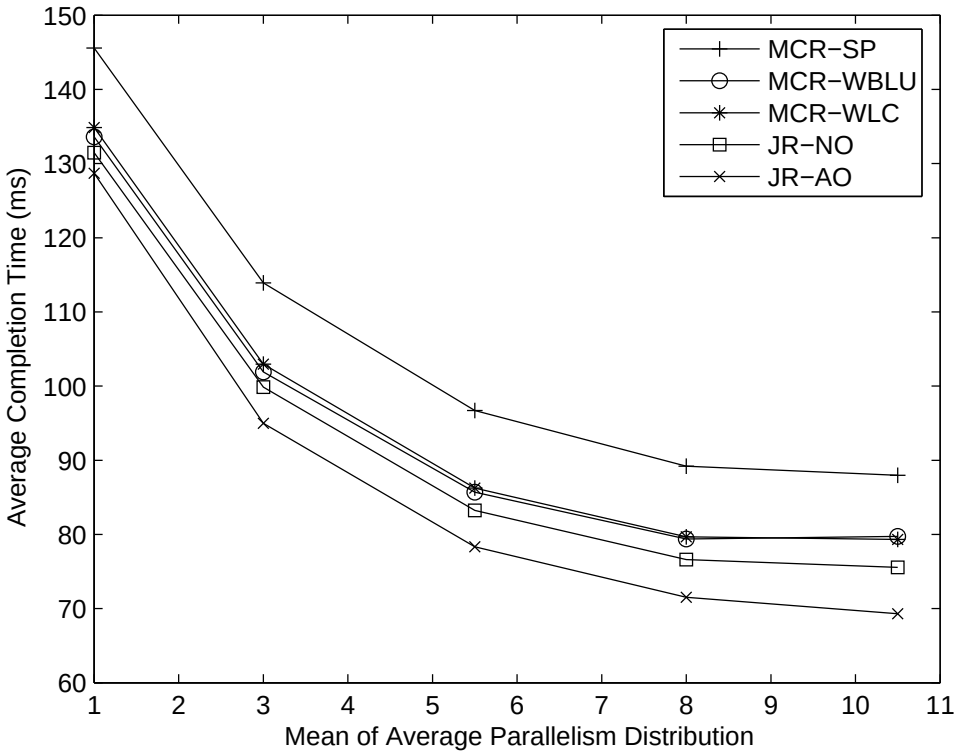

Figure 4.16: Graph of average completion time vs. mean of average parallelism

20 in the previous simulations. In this case, the maximum value of the uniform distribution is increased from 1 to 20, that is, the mean of the distribution function increased from 1 to 10.5. Reduction of the parallelism increases execution times of grid jobs because using multiple processors for a single job reduces completion time.

Another parameter in the grid job model is σ which is the parallelism variance coefficient. The change of average completion time for increasing mean of parallelism variance coefficient is shown in Figure 4.17. As σ decreases, speedup increases according to the model given in Chapter 2 and, consequently, average completion time decreases.

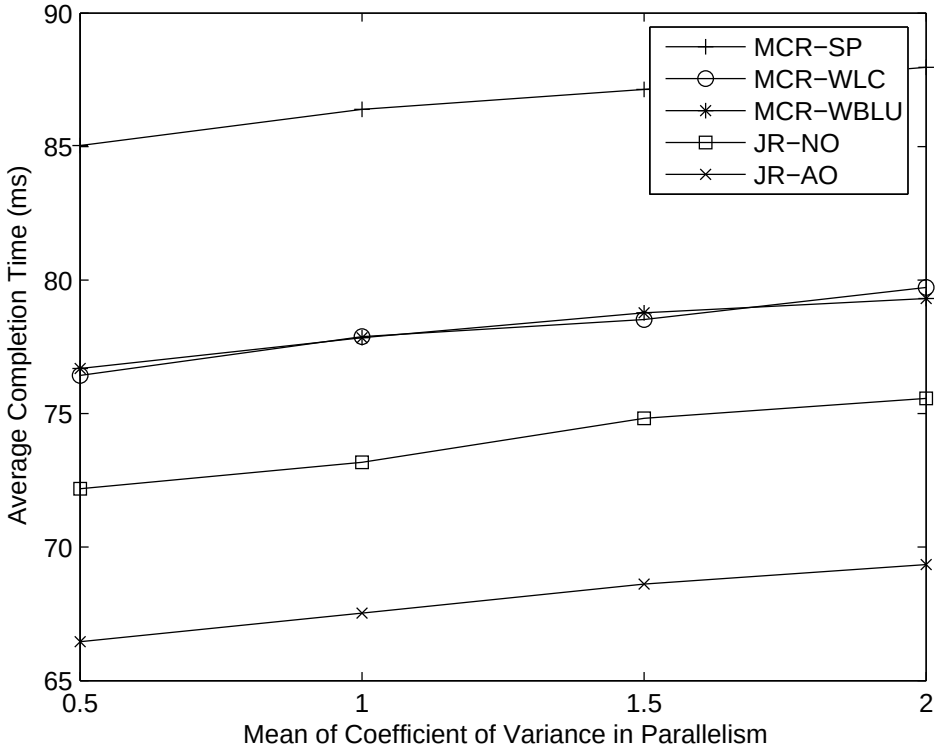

Figure 4.17: Graph of average completion time vs. mean of coefficient of variance in parallelism

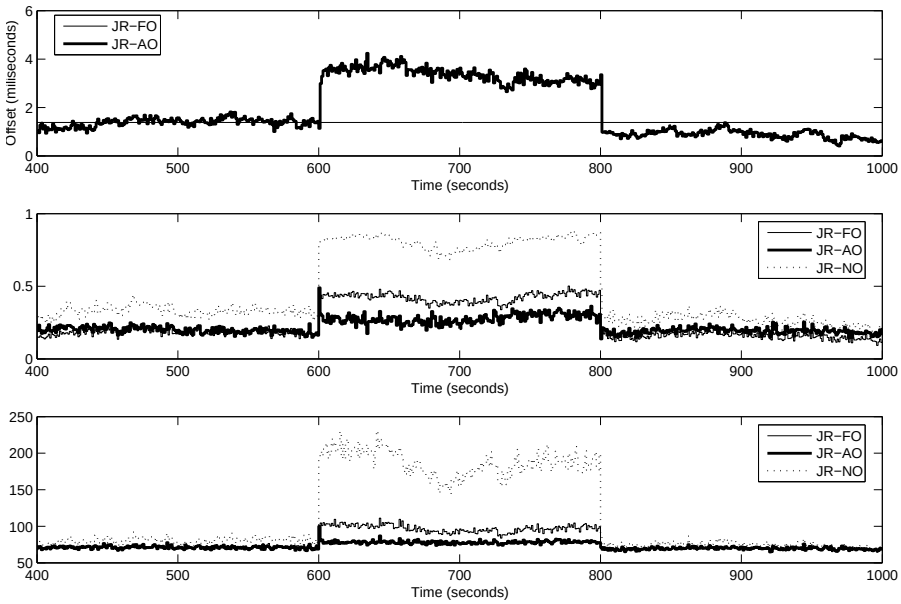

Figure 4.18: Graph of change in average extra offset, loss rate and average completion time for a sudden increase in the background load for $\gamma = 1$

4.5 Non-Stationary Traffic Scenarios

In reality, the average traffic load in network does not remain constant over time. The advantage of an adaptive congestion avoidance scheme is more significant in such a dynamic scenario because a fixed scheme cannot react changes in the network appropriately.

In this section, performance of the JR-AO, JR-FO and JR-NO is examined when the average background load is non-stationary. First, the reaction of the algorithms to a sudden increase in the background load is investigated. Later, their behavior in case of a sudden decrease in the load is presented.

4.5.1 Sudden Increase of Background Load

In Fig. 4.18, several performance metrics in case of a sudden increase of the background load can be seen. In this scenario, between 400 s and 600 s the

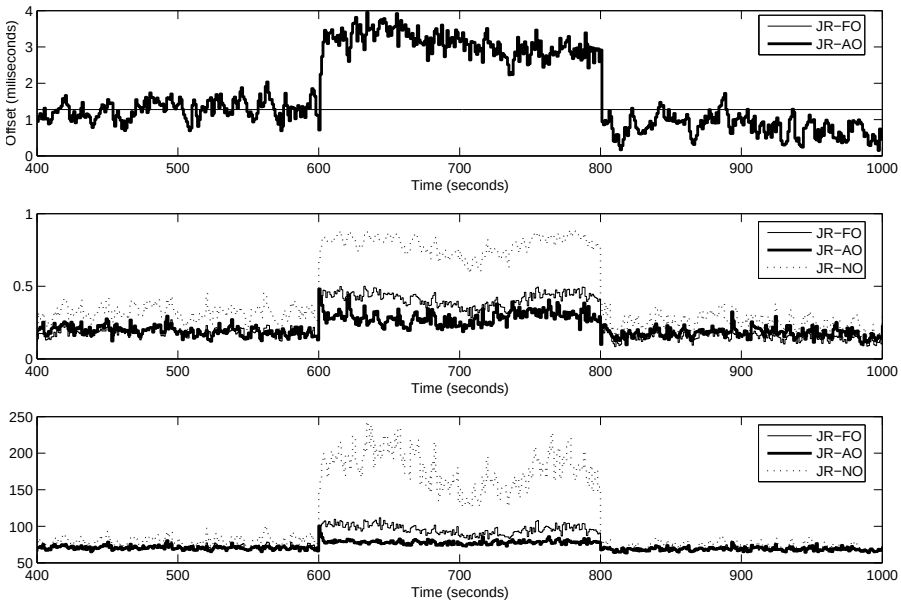

Figure 4.19: Graph of change in average extra offset, loss rate and average completion time for a sudden increase in the background load for $\gamma = 0.5$

average background load is 0.8 Erlangs and it is increased to 4 Erlangs at 600 s. Load is kept at that level until 800 s and, after that time, it is again reduced to 0.8 Erlangs. The first subplot shows the average offset value generated by JR-AO and the fixed offset value of JR-FO. The average offset generated by JR-AO in the low loss region is selected as the fixed offset value for JR-FO. The second subplot shows the change of average loss rate over time for JR-AO, JR-NO and JR-FO. Change of average completion time over time is shown in the third subplot.

From the figure, it can be seen that JR-AO reacts the increase in the background load by increasing the offset values for grid bursts and the benefit of this reaction can be observed in the loss rate and completion time graphs. There is a worsening in both metrics for all of the algorithms in the high load region but the disadvantage of JR-AO is less than the other algorithms. The average completion time is reduced 20% by JR-AO in the high load region in comparison to JR-FO and 60% in comparison to JR-NO.

Fig. 4.19 shows the results for the same scenario but in this case, burstiness factor $\gamma = 0.5$. Although an increase in the burstiness levels makes loss estimation difficult, JR-AO mechanism still outperforms JR-FO.

4.5.2 Sudden Decrease in the Background Load

A similar non-stationary scenario is a sudden decrease in the background load level. In this scenario, the load level is kept at 4 Erlangs between 400 s and 600 s. After that the background load is completely removed until 800 s. Later, it is increased to 4 Erlangs again. In this case, the fixed offset value is selected according to a high load situation.

From Fig. 4.20, it can be seen that applying extra offset increases completion time when the background load reduces to zero. From the loss rate graph, it can be seen that JR-FO achieves better loss rates than JR-AO. However, JR-FO has no or little advantage over JR-AO in terms of completion time in the high loss region.

In the low loss region, the reduction obtained by JR-AO is approximately 6 ms which is 8%. This amount is nearly the twice of the extra fixed offset which is unnecessary in the low load region. The reason of this doubling effect is that the extra offset is applied to both of job and job result bursts and the completion time of a grid job includes the extra offset of a job and job result result burst.

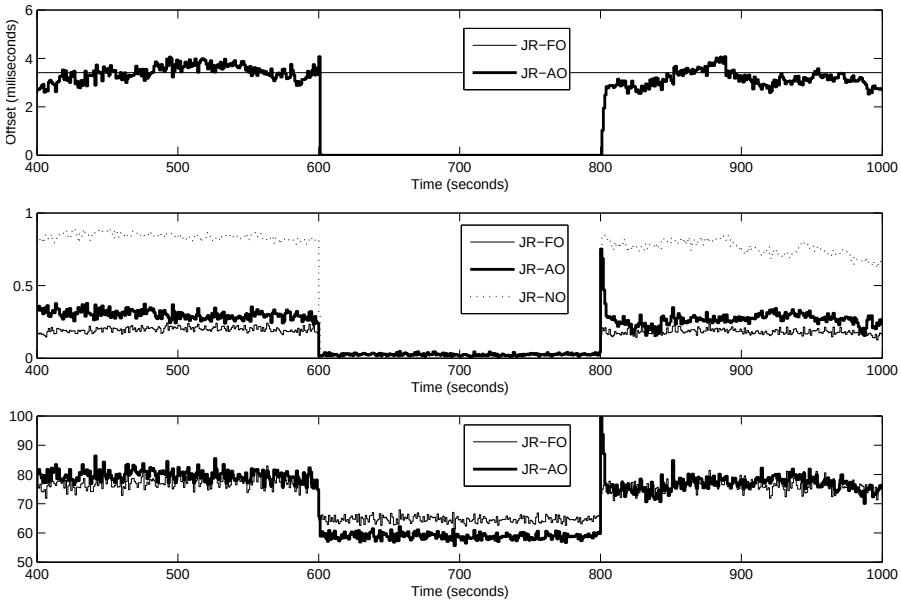

Figure 4.20: Graph of change in average extra offset, loss rate and average completion time for a sudden reduction in the background load for $\gamma = 1$

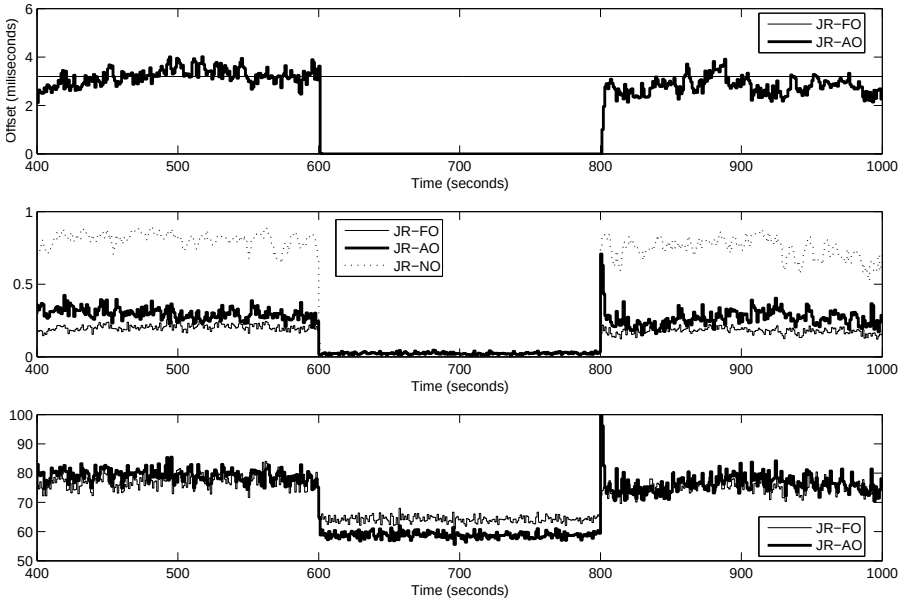

Figure 4.21: Graph of change in average extra offset, loss rate and average completion time for a sudden reduction in the background load for $\gamma = 0.5$

Chapter 5

CONCLUSIONS

In this thesis, we first propose mechanisms to collect feedback and notify burst losses for a consumer controlled OBS grid architecture. The feedback collection is required for congestion-based resource-path selection and burst loss notification is used to reduce grid job completion time in case of a loss. These mechanisms use minimal support from core routers and requires little signaling.

Next, the elements of the completion time of an OBS grid job are used to formulate the expected completion time function which includes the minimum required completion time and the expected retransmission delay caused by burst losses. A joint resource and path selection algorithm for grid consumers is proposed using this formulation. In this algorithm, the resource-path pair which minimizes this expected completion time is selected for grid burst transmission. A similar algorithm is also proposed for grid resources which selects the path to consumer.

It is shown that combining resource selection and path selection reduces congestion in the OBS grid network. Since OBS is not a reliable transmission protocol, retransmission of bursts become necessary when a burst is lost. This retransmission increases completion time of a consumer grid job and this delay

is critical for interactive applications. For that reason, network aware resource selection is very important for OBS grids especially when the load levels are high.

In addition to this joint resource-path selection algorithm, a QoS offset based service differentiation for grid bursts is presented in this thesis. This QoS algorithm computes a QoS offset minimizes the expected job completion time for grid bursts. QoS offset increases completion time because it increases transmission delay but the loss rate reduction achieved by the QoS offset may decrease overall completion time. The algorithm uses the analytical performance model of QoS offset to achieve a trade-off between the transmission delay and loss rate reduction. The algorithm can adapt to the load changes in the network, applying larger offset when the congestion is high and applying a smaller offset when the congestion is low, and it reduces completion times significantly.

Bibliography

[1] "Lhc - the large hadron collider: http://lhc.web.cern.ch/lhc/."

[2] "Seti@home: http://setiathome.ssl.berkeley.edu/."

[3] I. Foster and C. Kesselman, *The grid: blueprint for a new computing infrastructure*. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 1998.

[4] I. Taylor, R. Philp, M. S. Shields, O. F. Rana, and B. F. Schutz, "The consumer grid," (Toronto, Ontario, Canada), Global Grid Forum, February 2002.

[5] B. Volckaert, P. Thysebaert, M. D. Leenheer, F. D. Turck, B. Dhoedt, and P. Demeester, "Grid computing: The next network challenge!," *The Journal of The Communications Network*, vol. 3, pp. 159–168, 2004.

[6] D. Simeonidou, R. Nejabati, *et al.*, "Optical Network Infrastructure for Grid," *Global Grid Forum*, 2004.

[7] L. Renambot, T. van der Schaaf, H. Bal, D. Germans, and H. Spoelder, "Griz: experience with remote visualization over an optical grid," *Future Generation Computer Systems*, vol. 19, no. 6, pp. 871–882, 2003.

[8] C. Qiao, "Optical burst switching (OBS)–a new paradigm for an Optical Internet," *Journal of High Speed Networks*, vol. 8, no. 1, pp. 69–84, 1999.

[9] D. Simeonidou and R. Nejabati, "Grid Optical Burst Switched Networks (GOBS)," *Global Grid Forum Draft, May*, 2005.

[10] Y. Chen, C. Qiao, and X. Yu, "Optical burst switching: a new area in optical networking research," *Network, IEEE*, vol. 18, no. 3, pp. 16–23, 2004.

[11] N. Akar and E. Karasan, "Exact calculation of blocking probabilities for bufferless optical burst switched links with partial wavelength conversion," *Broadband Networks, First International Conference on*, pp. 110–117, 2004.

[12] C. Hsu, T. Liu, and N. Huang, "Performance analysis of deflection routing in optical burst-switched networks," *INFOCOM, IEEE*, vol. 1, 2002.

[13] X. Lu and B. Mark, "Performance modeling of optical-burst switching with fiber delay lines," *Communications, IEEE Transactions on*, vol. 52, no. 12, pp. 2175–2183, 2004.

[14] V. Vokkarane, J. Jue, and S. Sitaraman, "Burst segmentation: an approach for reducing packet loss in optical burst switched networks," *Communications, 2002. ICC 2002. IEEE International Conference on*, vol. 5, 2002.

[15] R. Ramaswami and K. Sivarajan, "Optimal routing and wavelength assignment in all-optical networks," *INFOCOM, IEEE*, pp. 970–979, 1994.

[16] E. Karasan and E. Ayanoglu, "Effects of wavelength routing and selection algorithms onwavelength conversion gain in WDM optical networks," *Networking, IEEE/ACM Transactions on*, vol. 6, no. 2, pp. 186–196, 1998.

[17] J. Li and C. Qiao, "Schedule burst proactively for optical burst switched networks," *Computer Networks*, vol. 44, no. 5, pp. 617–629, 2004.

[18] S. Wang, "Using TCP congestion control to improve the performances of optical burst switched networks," *Communications, IEEE International Conference on*, vol. 2, 2003.

[19] J. Teng and G. Rouskas, "Routing path optimization in optical burst switched networks," *Optical Network Design and Modeling, 2005. Conference on*, pp. 1–10, 2005.

[20] G. Thodime, V. Vokkarane, and J. Jue, "Dynamic congestion-based load balanced routing in optical burst-switched networks," in *Global Telecommunications Conference, IEEE*, pp. 2628–2632, December 2003.

[21] L. Yang and G. Rouskas, "Adaptive Path Selection in OBS Networks," *Journal of Lightwave Technology*, vol. 24, no. 8, p. 3002, 2006.

[22] S. Xu, T. Guerin, R. Kurose, J. Towsley, and D. Zhang, "Exploring the performance benefits of end-to-end path switching," *Network Protocols, IEEE International Conference on*, pp. 304–315, 2004.

[23] S. Tao, K. Xu, A. Estepa, T. Fei, L. Gao, R. Guerin, J. Kurose, D. Towsley, and Z. Zhang, "Improving VoIP quality through path switching," *INFOCOM, IEEE*, 2005.

[24] C. Gauger, "Trends in optical burst switching," *Proceedings of SPIE*, vol. 5247, pp. 115–125, 2003.

[25] M. Yoo and C. Qiao, "A new optical burst switching protocol for supporting quality of service," *SPIE Proceedings, All Optical Networking: Architecture, Control and Management Issues*, vol. 3531, pp. 396–405, 1998.

[26] M. Yoo and C. Qiao, "Supporting multiple classes of services in IP over WDM networks," *Global Telecommunications Conference*, vol. 1, 1999.

[27] M. Yoo, C. Qiao, and S. Dixit, "QoS performance of optical burst switching in IP-over-WDM networks," *Selected Areas in Communications, IEEE Journal on*, vol. 18, no. 10, pp. 2062–2071, 2000.

[28] K. Dolzer, C. Gauger, J. Späth, and S. Bodamer, "Evaluation of reservation mechanisms for optical burst switching," *AEU International Journal of Electronics and Communications*, vol. 55, no. 1, pp. 18–26, 2001.

[29] N. Barakat and E. Sargent, "An accurate model for evaluating blocking probabilities in multi-class OBS systems," *Communications Letters, IEEE*, vol. 8, no. 2, pp. 119–121, 2004.

[30] N. Barakat and E. Sargent, "Analytical Modeling of Offset-Induced Priority in Multiclass OBS Networks," *Communications, IEEE Transactions on*, vol. 53, no. 8, pp. 1343–1352, 2005.

[31] Y. Chen, M. Hamdi, and D. Tsang, "Proportional QoS over OBS networks," *Global Telecommunications Conference, IEEE*, vol. 3, 2001.

[32] V. Vokkarane, K. Haridoss, and J. Jue, "Threshold-based burst assembly policies for QoS support in optical burst-switched networks," *Proceedings, SPIE Optical Networking and Communication Conference (OptiComm) 2002*, pp. 125–136.

[33] V. Vokkarane and J. Jue, "Prioritized burst segmentation and composite burst-assembly techniques for QoS support in optical burst-switched networks," *Selected Areas in Communications, IEEE Journal on*, vol. 21, no. 7, pp. 1198–1209, 2003.

[34] W. Liao and C. Loi, "Providing service differentiation for optical-burst-switched networks," *Lightwave Technology, Journal of*, vol. 22, no. 7, pp. 1651–1660, 2004.

[35] H. Boyraz and N. Akar, "Rate-controlled optical burst switching for both congestion avoidance and service differentiation," *Optical Switching and Networking*, vol. 2, no. 4, pp. 217–229, 2005.

[36] M. De Leenheer, E. Van Breusegem, R. Thysebaert, B. Volckaert, F. De Turck, B. Dhoedt, P. Demeester, D. Simeonidou, M. Mahoney, R. Nejabati, *et al.*, "An OBS-based grid architecture," *Global Telecommunications Conference Workshops, IEEE*, pp. 390–394.

[37] C. Partridge, T. Mendez, and W. Milliken, "Host Anycasting Service," *Request For Comments*, vol. 1546, 1993.

[38] E. Basturk, R. Engel, R. Haas, V. Peris, and D. Saha, "Using Network Layer Anycast for Load Distribution in the Internet," *Proc. Global Internet 98*, 1998.

[39] M. De Leenheer, F. Farahmand, K. Lu, T. Zhang, P. Thysebaert, B. Volckaert, F. De Turck, B. Dhoedt, P. Demeester, and J. Jue, "Anycast Algorithms Supporting Optical Burst Switched Grid Networks," *Proceedings of the International conference on Networking and Services*, 2006.

[40] R. Nejabati, G. Zervas, G. Dimitriades, and D. Simeonidou, "Programmable optical burst switched network: A novel infrastructure for Grid," *5th IEEE/ACM Int. Symp. Cluster Computing Grid (CCGrid)*, pp. 9–12, 2005.

[41] D. Simeonidou, R. Nejabati, G. Zervas, D. Klonidis, A. Tzanakaki, and M. O'Mahony, "Dynamic optical network architectures and technologies for existing and emerging grid services," *Lightwave Technology, Journal of*, vol. 23, pp. 3347–3357, October 2005.

[42] D. Tennenhouse, J. Smith, W. Sincoskie, D. Wetherall, and G. Minden, "A survey of active network research," *Communications Magazine, IEEE*, vol. 35, no. 1, pp. 80–86, 1997.

[43] M. De Leenheer, P. Thysebaert, B. Volckaert, F. De Turck, B. Dhoedt, P. Demeester, D. Simeonidou, R. Nejabati, G. Zervas, D. Klonidis, and M. J. O'Mahony, "A view on enabling-consumer oriented grids through optical

burst switching," *Communications Magazine, IEEE*, vol. 44, pp. 124–131, March 2006.

[44] A. Iosup, C. Dumitrescu, D. Epema, H. Li, and L. Wolters, "How are real grids used? the analysis of four grid traces and its implications," *The 7th IEEE/ACM International Conference on Grid Computing (Grid)*, pp. 28–29.

[45] W. Cirne and F. Berman, "A model for moldable supercomputer jobs," *Parallel and Distributed Processing Symposium., Proceedings 15th International*, p. 8, 2001.

[46] A. B. Downey, "A parallel workload model and its implications for processor allocation," *Cluster Computing*, vol. 1, no. 1, pp. 133–145, 1998.

[47] R. Bhandari, *Survivable Networks: Algorithms for Diverse Routing*. Kluwer Academic Publishers, 2001.